کمپیوټر ساینس پوهنځی

د کمپیوټر جوړښت او اسمبلي ژبه

پوهندوی بادام نیازی

۱۴۰۲

پلورل منع دی

د کمپیوټر جوړښت او اسمبلي ژبه

پوهندوی بادام نیازی

Afghanic

Faculty of Computer Science

Assist Prof Badam Niazai

Computer Architecture & Assembly Language

دا کتاب د کمپیوټر ساینس، معلوماتي سیسـتمونو او انجنیري ټیکنالوژۍ په برخه کې دزده کونکو لپاره د کمپیوټر هارډویر او اسمبلي ژبې د پیژندنې په توګه لیکل شوی.
دا کتاب لس څپرکي لري، سیګنلونه او د اعدادو سیستمونه، بولیین منطق او منطقي ګیټونه، منترمونه، مکسترمونه، د کارناف نقشـه او عمومي ګیټونه، ترکیبي منطق، هم مهاله مسـلسـل منطق، د کمپیوټر د جوړښت پېژندګلوي، حافظه، اسمبلي ژبه او ا. آر. ام. ARM لارښوونې لومړۍ او دوهمه برخه د دې کتاب له مهمو عنوانونه څخه دي.
دا کتاب د ډیجیټل لجیک ډیزاین مضمون د ډیټابیس او معلوماتي سیستم په څانګه کې د یو تخصصـي مضمون په توګه تدریس کېږي، په همدې اساس د یادې څانګې محصلینو او د مسلک مینوالو ته به ډېر مؤثر تمام شي.

ښاغلی بادام نیازی د ګل جان زوی د لغمان ولایت د علینګار ولسوالۍ د سنګر کلي اوسېدونکی دی. په ۲۰۰٦ کال د ننګرهار پوهنتون د کمپیوټر سـاینس پوهنځي څخه په کادر کې فارغ او په همدې کال ببرته په همدې پوهنځي کې د استاد په توګه په دنده و ګمارل شو.
په ۲۰۱۲ کال د بورس له لارې پېښور پوهنتون ته کامیاب او په ۲۰۱۵ کال کې د ماسترۍ څخه فارغ شو.
ښاغلی نیازي د استادي تر څنګ ٦ کاله دژغورنې نړیوالې موسسې د ختیځ زون د ښوونې او روزنې د مسوول په توګه هم دنده اجرا کړې ده. د ښاغلي نیازی ډېرې مقالې په بېلابېلو نړیوالو ژورنالونو او کنفرانسونو کې خپرې شوې دي.

Funded by
Kinderhilfe-Afghanistan

ISBN 978-9936-622-65-4

2023

Not for Sale

د کمپیوټر جوړښت او اسمبلۍ ژبه

پوهندوی بادام نیازی

Faculty of Computer Science
کمپیوټر ساینس پوهنځی

Funded by
Kinderhilfe-Afghanistan

Pashto PDF
2023

افغانیک
Afghanic

Computer Architecture & Assembly Language

Assist Prof Badam Niazai

Download:
www.ecampus-afghanistan.org

اقرأ باسم ربك الذي خلق

د کمپيوټر جوړښت او اسمبلي ژبه

پوهندوی بادام نيازی

لومړی چاپ

دغه کتاب په پي ډي ايف فارمټ کې په مله سي ډي کې هم لوستلی شئ:

د کتاب نوم	د کمپیوټر جوړښت او اسمبلۍ ژبه
ژباړن	پوهندوی بادام نیازی
خپرندوی	ننگرهار پوهنتون، د کمپیوټر ساینس پوهنځی
وېب پاڼه	www.nu.edu.af
د چاپ کال	۱۴۰۲، لومړی چاپ
چاپ شمېر	۱۰۰۰
مسلسل نمبر	۳۸۴
ډاونلوډ	www.ecampus-afghanistan.org

دا کتاب د افغان ماشومانو لپاره د جرمني کمېټې، په جرمني کې د Eroes کورنۍ یوې خیریه ټولنې لخوا تمویل شوی دی.
اداري او تخنیکي چارې یې د افغانیک لخوا ترسره شوي دي.
د کتاب د محتوا او لیکنې مسوولیت د کتاب په ژباړن او اړوند پوهنځي پورې اړه لري. مرسته کوونکي او تطبیق کوونکي ټولنې په دې اړه مسوولیت نه لري.

د تدریسي کتابونو د چاپولو لپاره له موږ سره اړیکه ونیسئ:
ډاکتر یحیی وردک، د لوړو زده کړو وزارت، کارته ۴، کابل
موبایل ۰۷۸۰۲۳۲۳۱۰، ۰۷۰۶۳۲۰۸۴۴
ایمېل info@ecampus-afghanistan.org

د چاپ ټول حقوق له ژباړن سره خوندي دي.

ای اس بي ان ۹۷۸-۹۹۳۶-۶۲۲-۶۵-۴

د درسي کتابونو چاپول

قدرمنو استادانو او ګرانو محصلينو!

د افغانستان په پوهنتونونو کې د درسي کتابونو کموالی او نشتوالی له لويو ستونزو څخه ګڼل کېږي. يو زيات شمېر استادان او محصلين نويو معلوماتو ته لاسرسی نه لري، په زاړه مېتود تدريس کوي او له هغو کتابونو او چپترونو څخه ګټه اخلي چې پخواني دي او په بازار کې په ټيټ کيفيت فوتوکاپي کېږي.

موږ د ۲۰۱۰ څخه تر ۲۰۲۳ کال پورې د ننګرهار، خوست، کندهار، هرات، بلخ، البيروني، کابل پوهنتون، د کابل طبي پوهنتون او د کابل پولي تخنيک پوهنتون لپاره ۳۸۹ عنوانه مختلف درسي کتابونه د طب، ساينس، انجنيري، اقتصاد، ژورنالېزم او کرهنې پوهنځيو لپاره چاپ کړي دي.

د يادونې وړ ده، چې دغه چاپ شوي کتابونه د هېواد ټولو اړوندو پوهنتونونو او يو زيات شمېر ادارو او موسساتو ته په وړيا توګه وېشل شوي دي. ټول چاپ شوي کتابونه له www.ecampus-afghanistan.org ويب پاڼې څخه ډانلوډ کولی شئ.

دا چارې په داسې حال کې ترسره کېږي چې د افغانستان د لوړو زده کړو وزارت د (۲۰۱۰-۲۰۱۴) کلونو په ملي ستراتيژيک پلان کې راغلي دي چې:

"د لوړو زده کړو او د ښوونې د ښه کيفيت او زده کوونکو ته د نويو، کره او علمي معلوماتو د برابرولو لپاره اړينه ده، چې په دري او پښتو ژبو د درسي کتابونو د ليکلو فرصت برابر شي، د تعليمي نصاب د ريفورم لپاره له انګريزي ژبې څخه دري او پښتو ژبو ته د کتابونو او درسي موادو ژباړل اړين دي، له دغو امکاناتو پرته د پوهنتونونو محصلين او استادان عصري، نويو، تازه او کره معلوماتو ته لاسرسی نه شي پيدا کولای."

موږ غواړو چې د درسي کتابونو په برابرولو سره د هېواد له پوهنتونونو سره مرسته وکړو او د چپټر او لکچرنوټ دوران ته د پای ټکی کېږدو. د دې لپاره اړينه ده چې د افغانستان د پوهنتونونو لپاره هر کال لږ تر لږه ۱۰۰ عنوانه درسي کتابونه چاپ شي.

له ټولو درنو استادانو څخه هیله کوو، چې په خپلو مسلکي برخو کې نوي کتابونه ولیکي، ویې ژباړي او یا هم خپل پخواني لیکل شوي کتابونه، لکچرنوټونه او چپټرونه ایډیټ او د چاپ لپاره تیار کړي، زموږ په واک کې یې راکړي چې په ښه کیفیت چاپ او وروسته یې د اړوند پوهنځیو، استادانو او محصلینو ته په واک کې ورکړو. همدارنګه د یادو ټکو په اړه خپل وړاندیزونه او نظریات له موږ سره شریک کړي، چې په ګډه په دې برخه کې اغېزمن ګامونه پورته کړو.

د لیکوالانو او خپرونکو له خوا پوره زیار ایستل شوی، چې د کتابونو محتویات د نړیوالو علمي معیارونو پر اساس برابر شي، خو بیا هم کېدای شي د کتاب په محتوا کې ځینې تېروتنې او ستونزې ولیدل شي، نو له درنو لوستونکو څخه هیله لرو چې خپل نظریات او نیوکې لیکوال او یا موږ ته په لیکلې بڼه راولېږي، چې په راتلونکي چاپ کې اصلاح شي.

د افغان ماشومانو لپاره د جرمني کمېټې او د هغې له مشر ډاکتر ایروس څخه ډېره مننه کوو چې د دغه کتاب د چاپ لګښت یې ورکړی دی. دوی تر دې مهاله د ننګرهار پوهنتون د ۲۵۰ عنوانه طبي او غیر طبي کتابونو د چاپ لګښت پر غاړه اخیستی دی.

د پوهنتونونو رئیسانو، د پوهنځیو رئیسانو او استادانو څخه مننه کوم چې د کتابونو د چاپ لړۍ یې هڅولې او مرسته یې ورسره کړې ده. د دغه کتاب له لیکوال څخه ډېر منندوی یم او ستاینه یې کوم، چې د کلونو - کلونو زیار محصول یې په وړیا توګه ګرانو محصلینو ته وړاندې کړ.

همدارنګه د خپل دفتر له همکارانو هر یو ښاغلي حکمت الله عزیز، ښاغلي فهیم حبیبي، ښاغلي ګل آغا احمدي او ښاغلي هېواد صافی څخه هم مننه کوم، چې د کتابونو د چاپ په برخه کې یې نه ستړې کېدونکې هلې ځلې کړې دي.

ډاکتر یحیی وردک
د لوړو زده کړو وزارت، کابل، جون، ۲۰۲۳
د دفتر ټیلیفون: ۰۷۸۰۲۳۲۳۱۰، ۰۷۰۶۳۲۰۸۴۴
ایمیل: info@ecampus-afghanistan.org

د ژباړن خبرې

له لوی الله تعالی څخه شکر ادا کوم چې ما ته یې توفیق راکړ تر څو د خپل مسلک اړوند یو اړین درسي کتاب خپلې ملي ژبې (پښتو) ته وژباړم او له دې لارې خپل هیواد او هیوادوالو ته د خدمت مصدر وګرځم.

ژباړل شوی کتاب چې د کمپیوټر سیستمونه ډیجیټل ډیزاین، د کمپیوټر جوړښت او اسمبلی ژبې اساسات تر سرلیک لاندې دعطا لخوا تالیف شوی، یاد کتاب نړیوال شهرت لري او د متحده ایالاتو د نوي هاروین د ایالت په غربي کنیکټې کټ مرکزي پوهنتون د کمپیوټر ساینس په څانګه کې تدریس او په ۲۰۱۸ زیږدیز کال کې چاپ شویدی. د ننګرهار پوهنتون د کمپیوټر ساینس پوهنځی د ډیټابیس او معلوماتی سیستم د څانګې له خوا ماته د پوهنمل علمي رتبې څخه پوهندوی علمي رتبې ته د ترفیع په موخه ملي (پښتو) ژبې ته د ژباړل کیدو له پاره د شهید استاد ربانی د ښوونې او روزنې پوهنتون د کمپیوټر ساینس پوهنځی محترم استاد پوهنوال صدیق الله بارکزې تر مستقیمې لارښوونې لاندې راکړل شوی ؤ چې د لوړو زده کړو وزارت د انسجام لوی ریاست د استادانو د علمي موضوعاتو ټاکلو کمیسون په ۱۳۹۸/۱۰/۱۷ نیټه غونډه کې مطرح او په (۶۴۳) ګڼه تحقیق، تالیف اوژباړې په ډیټابیس کې ثبت کړه.

ما هم تر خپل توان پورې هڅه کړیده چې د ژباړې د اصولو مطابق د کتاب او لیکوال حقوق خوندي وساتم او په پوره امانتدارۍ سره پرته د تورو زیاتولو او کمولو څخه مې خپل مورنۍ ژبې (پښتو) ته ژباړلی دی.

دا چې د ډیجیټل لاجیک ډیزاین مضمون د ډیټابیس او معلوماتي سیستم په څانګه کې د یو تخصصی مضمون په توګه تدریس کیږي، له نیکه مرغه ژباړل شوی کتاب له ۸۰ سلنه برخه په بر کې نیولي . په همدې اساس د یادې څانګې محصلینو او د مسلک مینوالو ته به ډیر مؤثر تمام شي.

په اخر کې له خپل لارښود استاد محترم پوهنوال صدیق الله بارکزی او ټولو هغو دوستانو څخه د زړه له تله مننه کوم چې ماسره یې د دې کتاب په ژباړلو کې خپل قیمتی وخت مصرف کړی او اړیني مشورې یې راکړي دي.

درنښت

بادام نیازی

لیکلړ

لومړی څپرکی
(سیګنلونه او د اعدادو سیستمونه)

سیګنلونه او د اعدادو سیستمونه ۲

۱.۱ پېژندنه ۲

۲.۱ انالوګ سېګنل (Analog Signal) ٤

۱.۲.۱ د انالوګ سېګنل ځانګړنې ٥

(Characteristics Of An Analog Signal): د مهالي انالوګ سېګنل ځانګړتیاوې عبارت دي له فریکونسي، امپلیتود او پړاو (Phase) څخه. ٥

۳.۱ ډیجیټل سېګنلونه (Digital Signals) ۸

۴.۱ د اعدادو سیستم (Number System) ۸

۱.۴.۱ د دوو له قاعدې څخه د لسو قاعدې ته تبدیلول (Converting From Binary To Decimal): ۹

۳.۴.۱ د لسو قاعدې د کسري (اعشاري) اعدادو تبدیلول د دوو قاعدې ته (Converting Decimal Fraction To Binary)..................... ۱۲

۴.۴.۱ د شپاړسو قاعدې اعدادو تبدیلول د دوو قاعدې اعدادو ته (Converting From Hex to Binary) ۱٤

۵.۴.۱ د دوه په قاعدې اعدادو جمع کول(Binary Addition)...... ۱۵

۵.۱ کمپلیمنټ (Complement) او دوه ییز کمپلیمنټ (Two's Complement) ۱۵

۱.۵.۱ د دوه غیرې علامه لرونکو عددونو تفریق کول د دوه ییز کمپلیمنټ په طریقه(Subtraction of Unsigned Number Using Two's Complement) ۱٦

۶.۱ بې علامې ، علامه لرونکي، میګنیچیوډ، او د دوه په قاعده دوه ییز کمپلیمنټ

عدد (Unsigned, Signed Magnitude, and Signed Two's
Complement Binary Numbe) ۱۸
۷.۱ د دوو په قاعدې د اعدادو جمع کول د علامه لرونکي دوه ییز کمپلیمنټ په ذریعه (Binary Addition Using Signed Two's Complement) ۱۹
۸.۱ د اعشاري اعدادو ښودل ۲۱
(Floating Point Representation) ۲۱
۱.۸.۱ یوګونې دقیقه ښودنه (Single Precision Representation) ۲۲
۹.۱ د دوو په قاعدې د لسو قاعده ییز اعداد (Binary-Coded Decimal (BCD)) ۲٤
۱۰.۱ د کوډ ورکولو سیستم (Coding Schemes) ۲۵
۱.۱۰.۱ اسکي کوډ (ASCII Code) ۲۵
۲.۱۰.۱ یونیکوډ (Unicode) یا نړیوال کوډ ورکولو سیستم (Universal Code or Unicode) ۲۵
۱.۱۱ پیریتي بیټ (Parity Bit) ۲۹
۱۳.۱ د لیږلو طریقې (Transmission Modes) ۳۱
۱.۱۳.۱ وقفه یي لیږد (Asynchronous Transmission) ۳۱
۲.۱۳.۱ مسلسل لیږد (Synchronous Transmission) ۳۲
۱۴.۱ د لیږدونې طریقې (Transmission Methods) ۳۳
۱.۱۴.۱ مسلسل لیږد (Serial Transmission) ۳٤
۲.۱۴.۱ موازی لیږد (Parallel Transmission) ۳٤
۱۵.۱ خلاصه ۳۵

دوهم څپرکی

(بولیین منطق او منطقي ګیټونه)

بولیین منطق او منطقي ګیټونه ٤١

١.٢ پیژندنه ٤٢

٢.٢ بولین منطق او منطقي ګیټونه ٤٢

(Boolean logic and logic gates) ٤٢

٣.٢ مدغم شوي سرکټ (Integrated circuit) طبقه بندي ٥٠

٤.٢ بولین الجبرا تیورمس (Boolean Algebra Theorems) ٥٣

٥.٢ بولیین (Boolean) تابع (function)... ٥٨

٦.٢ خلاصه ٦٠

دریم څپرکی

(منترمونه، مکسترمونه، د کارناف نقشه او عمومي ګیټونه)

منترمونه (Minterms)، مکسترمونه (Maxterms)، ٦٦

د کارناف نقشه (Karnaugh Map)، او عمومي ګیټونه (Universal Gates) ٦٦

١.٣ پیژندنه ٦٧

٢.٣ منترمونه (Minterms) ٦٧

٣.٣ مکسترمونه (Maxterms).. ٧٢

٤.٣ کارناف نقشه (K-Map) ٧٣

١.٤.٣ درې –متحوله نقشه (Three-Variable Map) ٧٦

٢.٤.٣ څلور –متحوله نقشه (Four-Variable K-Map) ٨٠

٥.٣ د ضرابیو مجموعه (SOP) او د مجموعې ضرب (POS)........................ ٨٣

(Sum Of Products (Sop) And Product Of Sums (POS)) ٨٣

۶.۳ د نه پاملرنې حالتونه (Don’t Care Conditions).......................... ۸۵
۷.۳ عمومی گېټونه (Universal Gates).. ۸۷
۱.۷.۳ د نانډ (NAND) گېټ استعمال:.. ۸۷
۲.۷.۳ د نار (NOR) گېټ استعمال:.. ۸۸
۳.۷.۳ د منطقي تابع گانو یوازی تطبیق کول د نانډ (NAND) یا نار (NOR) گېټونو په کارولو سره (Implementation Of Logic Functions (Using NAND Gates Or NOR Gates Only ۹۰
۴.۷.۳ د نانډ (NAND) گېټ استعمال: ۹۱
۵.۷.۳ د نار(NOR) گېټ استعمال:.. ۹۲

خلورم څپرکی

(ترکیبي منطق Combinational Logic)

ترکیبي منطق(Combinational Logic) ۹۹
۲.۴ د ترکیبي منطق تحلیل (Analysis of Combinational Logic)... ۱۰۱
۳.۴ د ترکیبي منطق ډیزاین ۱۰۱
(Design of Combinational Logic).. ۱۰۱
۴.۴ اصلي بڼې ته بدلوونکی (Decoder) ۱۰۶
۵.۴ بل حالت ته بدلوونکې (Encoder).. ۱۰۷
۶.۴ ملټیپلیکسر (Multiplexer) (MUX) ۱۰۹
۱.۶.۴ د کوچنی ملټیپلکسر څخه په استفادی د لوی ملټیپلکسر ډیزاین کول ۱۱٤
(Designing Large Multiplexer Using Smaller Multiplexers) ۱۱٤
۲.۶.۴ د ملټیپلکسر څخه په استفادې د تابع اجرا کول ۱۱٦

(Implementing Functions Using Multiplexer).................... ۱۱٦
۷.۴ نیم جمع کوونکی ، پوره جمع کوونکی ،دوه یز جمع کوونکی او تفریق کوونکی
.... ۱۱۸
(Half Adder, Full Adder, Binary Adder, and Subtractor)... ۱۱۸
۱.۷.۴ څلور بیټه دوه یزه جمع کوونکی(4-Bit Binary Adder) ۱۲۱
۲.۷.۴ تفریق کوونکی (Subtractor) ۱۲۳
۸.۴ حسابي منطقي برخه (Arithmetic Logic Unit)....................... ۱۲۳
۴,۹ اوو خطو په واسطه د اعداو ښودنه (Seven-Segment Display) ۱۲۵
۱۰.۴ خلاصه ۱۲۹

پنځم څپرکی

(هم مهاله مسلسل منطق)

هم مهاله مسلسل منطق ۱۳۵
۱.۵ پیژندنه ۱۳٦
۲.۵ اس-ار لیچ (S-R Latch).. ۱۳۸
۳.۵ ډی فلیپ پلاب (D Flip-Flop).. ۱۳۹
۴.۵ J-K فلیپ پلاپ (J-K Flip-Flop) ۱٤۱
۵.۵ ټی فلیپ فلاپ (T Flip-Flop).. ۱٤۲
٦.۵ راجستر (registeR) ۱٤۲
۷.۵ د فریکونسي تجزیه کوونکی د J-K Flip-Flop په استعمال سره
(Frequency Divider Using J-K Flip-Flop)............................ ۱٤٥
۸.۵ د ترتیبي منطق تحلیل (Analysis Of Sequential Logic)............ ۱٤۷
۹.۵ ستیټ ډیاګرام (State Diagram).. ۱٥۰
۱۰.۵ فلیپ فلاپ متحرک جدول (Flip-Flop Excitation Table) ۱٥۱
د Excitation Table عملیي ۱٥۲

۱۱.۵ شمیرونکی یا Counter ۱۵٤
۱۲.۵ خلاصه ۱۵۷
پوښتنې ۱۵۹

شپږم څپرکی

(د کمپیوټر د جوړښت پیژندګلوي)

د کمپیوټر د جوړښت پیژندګلوي ۱٦۳
(Introduction To Computer Architecture) ۱٦۳
۱.٦ پیژندګلوي ۱٦٤
۲.٦ د کوچني کمپیوټر جوړښتیزې برخې (Components Of
Microcomputer) ۱٦٤
۱.۲.٦ د پروسیس مرکزي واحد (CPU) ۱٦٥
۲.۲.٦ د پروسیس مرکزي واحد وصلوونکې (CPU Buses) ۱٦٦
۳.۲.٦ د ۳۲ بیټه او ٦۴ بیټه پروسیس مرکزي واحد ترمنځ توپیر ۱٦۹
(Bit Versus 64-Bit CPU-۳۲) ۱٦۹
۳.٦ د مرکزي پروسېس کوونکی واحد ټکنالوژي (CPU Technology) ۱٦۹
۴.٦ د پروسیس مرکزي واحد جوړښت (CPU Architecture)................... ۱۷۲
۵.٦ د اینټیل کمپنی کوچني پروسیسر کورنی(Intel Microprocessor Family)
.... ۱۷۳
٦.٦ د څو هستو لرونکو پروسیسر (Multi-Cores Processors)............. ۱۷٦
۷.٦ د پروسیس مرکزي واحد په واسطه د لارښوونو تر سره کول (Cpu
Instruction Execution) ۱۷۷
۸.٦ د ډیسک اداره کوونکی (Disk Controller) ۱۸۱
۱.۹.٦ د صنعت معیاری جوړښت (Isa Bus) وصلونکی ۱۸۳

۲.۹.٦ د کوچني چینل جوړښت وصلونکې (MCA)........................ ۱۸۳
۳.۹.٦ د (EISA) وصلونکې (Bus).. ۱۸۳
۴.۹.٦ د ویزا (VESA) وصلونکې (Bus) ۱۸۳
۵.۹.٦ د پی سی ایي (PCI) وصلونکې (Bus) ۱۸٤
٦.۹.٦ یو ایس بی(Universal Serial Bus (USB ۱۸٥
۷.۹.٦ د یو ایس بی جوړښت (Usb Architecture) ۱۸٥
۸.۹.۶ د وسایلو ترمنځ وصلونکې PCI Express Bus ۱۸۸
۹.۹.۶ د وسایلو ترمنځ د وصولونکي جوړښت (PCI Express
Architecture)... ۱۸۹
۱۰.۹.٦ د PCI Express پروتوکول ۱۹۰
۱۰.٦ فایروال (Firewire) ۱۹۱
۱۱.٦ خلاصه ۱۹۳

اووم څپرکی
(حافظه Memory)

حافظه ۱۹۸
۱.۷ سریزه ۱۹۹
۲.۷ نیمه هادي حافظه (Semiconductor Memory) ۱۹۹
۳,۷ مرستندویه زېرمه (Hard Disk) ۲۰٦
۴,۷ ایس ایس ډي (SSD (Solid State Driver ۲۱۰
۵,۷ حافظې تسلسل (Memory Hierarchy) ۲۱۰
۱,۵,۷ خیالي حافظه (Virtual Memory)................................. ۲۲٤

۲,۵,۷ د کمپیوټر د حافظې جوړښت (Memory Organization Of A Computer) ۲۲۷

اتم څپرکی

(اسمبلي ژبه او ارم ARM لارښوونې لمړی برخه)

اسمبلي ژبه او ارم ARM لارښوونې لمړی برخه ۲۳۶

۸,۱ پېژندګلوي ۲۳۷

۲,۸ د لارښوونو د سېټ جوړښت (ISA) ۲۳۹

۱,۲,۸ د لارښوونو طبقه بندي د عمل وربانډی ترسره کیدونکو د تعداد له نظره (Classification of Instruction Based on Number of Operands) ۲۳۹

۳,۸ د ارم پروسیسر جوړښت (ARM Processor Architecture)......... ۲٤۱

۴,۸ د ارم راجسټرونه (ARM Registers).................................. ۲٤٤

۵,۸ د آرم لارښوونې (ARM Instructions)................................... ۲٤۷

۱,۵,۸ د ډېټا د پروسیس کولو لارښوونې (Data Processing Instructions) ۲٤۷

۲,۵,۸ مقایسوی او ازمایښتی لارښوونی (Compare And Test Instruction)... ۲۵۰

۳,۵,۸ د راجسترونو تبادله کولو لارښوونې (Register Swap Instruction (MOV,MVN)) ۲۵۲

۴,۵,۸ شیفټ او روټیټ لارښوونې (Shift And Rotate Instruction) ۲۵۳

۵,۵,۸ د آرم شرطیه او غیر شرطیه لارښوونې (ARM Unconditional Instructions And Conditional Instructions).................. ۲۵٦

٦,۸ د آرم د معلوماتو د پروسیس کوونکو لارښوونو جوړښت ۲۵۸

۷,۸ د ستیک عملیې او لارښوونې (Stack Operation And Instructions) ۲٦۱

۸,۸ برینچ او برینچ له لینک لارښوونوسره Branch (B) and Branch With (Link) Instruction ۲٦٤
۱/۸/۸ د B او BL لارښوونی جوړښت ۲٦٥
۹,۸ – *Multiply* (***MUL***) ***and Multiply*** ***Accumulate*** (***MLA***) ***Instructions***: ۲٦٦
۱,۹,۸ د ضربي لارښوونو جوړښت (Multiply Instruction Format) ۲٦٦
۱۰,۸ خلاصه ۲٦۷

نهم څپرکی

(د ارم (ARM) لارښوونې دوهمه برخه)

د ارم (ARM) لارښوونې دوهمه برخه ۲۷۱
۱,۹ پیژندنه ۲۷۲
۲,۹ د ای ار ایم د معلوماتو د لیږد لارښوونې (ARM Data Transfer (Instructions ۲۷۲
۱,۲,۹ د آی ارام مشابه لارښوونې (ARM Pseudo Instructions).. ۲۷۳
۹.۲.۲ د ذیرمه کولو لارښوونې (Store Instruction) ۲۷٤
۳,۹ د ای ار ام ادرس ورکولو طریقې (ARM Addressing Mode) ۲۷٤
۱,۳,۹ سمدستي ادرس ورکول (Immediate Addressing).......... ۲۷٦
۲,۳,۹ مخکې ترتیب شوی (Pre Indexed) ۲۷٦
۳,۳,۹ د بیرته لیکلو سره مخکې نه ترتیب شوی Pre-Indexed With (Write Back) ۲۷۸
۴,۳,۹ د وروسته ترتیب شوي ادرس ورکول (Post-index Addressing) ۲۷۹
۴,۹ د راجستر او حافظې تر منځ ادلون بدلون (Swap Memory And

(Register.. ۲۸۰
۵,۹ د بیټونو ساحوي لارښوونې (Bits Field Instructions)................. ۲۸۰
۱,۵,۹ ساحوی بیټونو روښانه لارښوونې (Bit Field Instruction(BFC)
.... ۲۸۰
۲,۵,۹ د بیټ ځایولو لارښوونې (Bit Filed Instruction)............. ۲۸۱
۶,۹ د معلوماتو ښکاره کول او حافظه ۲۸۲
(Data Representation And Memory) ۲۸۲
۷,۹ خلاصه ۲۸۵

لسم څپرکی

(کېل پروگرام لیکلو کاریال ای ار ام اسمبلي پروگرام ژبې له پاره)

۱,۱۰ پیژندنه ۲۹۰
۲,۱۰ کېل پروگرام لیکلو کاریال ای ار ام اسمبلي پروگرام ژبې له پاره ۲۹۱
۱,۲,۱۰ پروگرام بشپړونکی، د پروگرام راټولول (Assembling A
(Program ۲۹۵
۲,۲,۱۰ د ډیبگر (Debugger)/ سیمیلیټر (Simulator) چلول ۲۹۹
۳,۲,۱۰ د پروگرام خاکه (Program Template) ۳۰۳
۳,۱۰ د پروگرام لیکلو اصول (Programming Rules)..................... ۳۰٤
۴,۱۰ د معلوماتو څرگندول او حافظه (Data Representation and
(Memory ۳۰۵
۵,۱۰ لارښود (Directives) ۳۰٦
۱,۵,۱۰ د معلوماتو لارښود (Data Directive) ۳۰٦
۶/۱۰ حافظه په ویژن ۵ کې (Memory in µVision v5) ۳۰۸
ففبب (ffcC) ۳۱۰

ضمایم

ضمایم: ډیجیټل ډیزاین لابراتواري تجربي د LOGISIM په استعمال سره...... ۳۱۶

دوهم څپرکی ۳۲۰

ځوابونه ۳۲۰

شپږم څپرکی ۳۳۹

د بیا کتنې پوښتنې ۳۳۹

اووم څپرکی: پوښتنی او مشکلات ۳۴۲

اتم څپرکی ۳۴۸

ماخذونه ۳۶۰

انډکس ۳۶۲

د انځورونو لیکلړ

انځور ۱- ۱- ۲

انځور ۱- ۲- ۴

انځور ۱- ۳- ۵

انځور ۱- ۴- ۵

انځور ۱- ۵- ٦

انځور ۱- ٦- ۷

انځور ۱- ۷- ۸

انځور ۱- ۸- ۲۲

انځور ۱- ۹- ۲۳

انځور ۱- ۱۰- ۲۸

انځور ۱- ۱۱- ۳۰

انځور ۱- ۱۲- ۳۱

انځور ۱- ۱۳- ۳۲

انځور ۱- ۱۴- ۳۲

انځور ۱- ۱۵- ۳۳

انځور ۱- ۱٦- ۳۴

انځور ۲- ۱- ۴۴

انځور ۲- ۲- ۴۴

انځور ۲- ۳- ۴۵

انځور ۲- ۴- ۴۵

انځور ۲- ۵- ۴۵

انځور ۲- ٦- ۴٦

انځور ۲- ۷- ۴۷

انځور ۲- ۸- ۴۷

انځور ۲- ۹- ۴۷

انځور ۲- ۱۰- ۴۸

انځور ۲- ۱۱- ۴۹

انځور ۲- ۱۲- ۵۰

انځور ۲- ۱۳- ۵۰

انځور ۲- ۱۴- ۵۱

انځور ۲- ۱۵- ۵۱

انځور ۲- ۱٦- ۵۱

انځور ۲- ۱۷- ۵۲

انځور ۲- ۱۸- ۵۲

انځور ۲- ۱۹- ۵۸

انځور ۲- ۲۰- ۵۸

انځور ۳- ۱- ۷۱

انځور ۳- ۲- ۷۵

انځور ۳- ۳- ۷۵

انځور ۳- ۴- ۷٦

انځور ۳- ۵- ۷۷

انځور ۳- ٦- ۷۸

انځور ۳- ۷- ۷۸

انځور ۳- ۸- ۷۸

انځور ۳- ۹- ۸۰

انځور ۳- ۱۰- ۸۱

انځور ۳- ۱۱- ۸۲
انځور ۳- ۱۲- ۸۳
انځور ۳- ۱۳- ۸۳
انځور ۳- ۱۴- ۸۵
انځور ۳- ۱۵- ۸۵
انځور ۳- ۱٦- ۸٦
انځور ۳- ۱۷- ۸٦
انځور ۳- ۱۸- ۹۰
انځور ۳- ۱۹- ۹۱
انځور ۳- ۲۰- ۹۲
انځور ۴- ۱- ۱۰۰
انځور ۴- ۲- ۱۰۰
انځور ۴- ۳- ۱۰۲
انځور ۴- ۴- ۱۰۲
انځور ۴- ۵- ۱۰۴
انځور ۴- ٦- ۱۰۴
انځور ۴- ۷- ۱۰۴
انځور ۴- ۸- ۱۰۵
انځور ۴- ۹- ۱۰۵
انځور ۴- ۱۰- ۱۰٦
انځور ۴- ۱۱- ۱۰٦
انځور ۴- ۱۲- ۱۰۸
انځور ۴- ۱۳- ۱۰۹
انځور ۴- ۱۴- ۱۰۹

انځور ۴- ۱۵- ۱۱۰
انځور ۴- ۱٦- ۱۱۰
انځور ۴- ۱۷- ۱۱۲
انځور ۴- ۱۸- ۱۱۲
انځور ۴- ۱۹- ۱۱۳
انځور ۴- ۲۰- ۱۱۴
انځور ۴- ۲۱- ۱۱۵
انځور ۴- ۲۲- ۱۱۵
انځور ۴- ۲۳- ۱۱۷
انځور ۴- ۲۴- ۱۱۸
انځور ۴- ۲۵- ۱۱۹
انځور ۴- ۲٦- ۱۱۹
انځور ۴- ۲۷- ۱۲۰
انځور ۴- ۲۸- ۱۲۰
انځور ۴- ۲۹- ۱۲۱
انځور ۴- ۳۰- ۱۲۲
انځور ۴- ۳۱- ۱۲۲
انځور ۴- ۳۲- ۱۲۴
انځور ۴- ۳۳- ۱۲۴
انځور ۴- ۳۴- ۱۲۵
انځور ۴- ۳۵- ۱۲٦
انځور ۴- ۳٦- ۱۲٦
انځور ۴- ۳۷- ۱۲٦
انځور ۴- ۳۸- ۱۲۷

انځور ۴- ۳۹- ۱۲۸
انځور ۵- ۱- ۱۳۷
انځور ۵- ۲- ۱۳۷
انځور ۵- ۳- ۱۳۷
انځور ۵- ۴- ۱۳۷
انځور ۵- ٦- ۱۴۰
انځور ۵- ۷- ۱۴۰
انځور ۵- ۸- ۱۴۱
انځور ۵- ۹- ۱۴۳
انځور ۵- ۱۰- ۱۴۴
انځور ۵- ۱۱- ۱۴۴
انځور ۵- ۱۲- ۱۴۵
انځور ۵- ۱۳- ۱۴۵
انځور ۵- ۱۴- ۱۴٦
انځور ۵- ۱۵- ۱۴٦
انځور ۵- ۱٦- ۱۴۸
انځور ۵- ۱۷- ۱۴۹
انځور ۵- ۱۸- ۱۴۹
انځور ۵- ۱۹- ۱۴۹
انځور ۵- ۲۰- ۱۵۰
انځور ۵- ۲۱- ۱۵٦
انځور ۵- ۲۲- ۱۵۷
انځور ٦- ۱- ۱٦۵
انځور ٦- ۲- ۱٦٦

انځور ٦- ٣- ١٦٦
انځور ٦- ۴- ١٧٢
انځور ٦- ۵- ١٧٢
انځور ٦- ٦- ١٧۵
انځور ٦- ٧- ١٧٦
انځور ٦- ٨- ١٧٧
انځور ٦- ٩- ١٧٨
انځور ٦- ١٠- ١٧٨
انځور ٦- ١١- ١٨٠
انځور ٦- ١٢- ١٨٢
انځور ٦- ١٣- ١٨٢
انځور ٦- ١۴- ١٨۴
انځور ٦- ١۵- ١٨٧
انځور ٦- ١٦- ١٨٧
انځور ٦- ١٧- ١٨٩
انځور ٦- ١٨- ١٩٠
انځور ٦- ١٩- ١٩٠
انځور ٦- ٢٠- ١٩١
انځور ٦- ٢١- ١٩٢
انځور ٦- ٢٢- ١٩۴
انځور ٧- ١- ٢٠٠
انځور ٧- ٢- ٢٠٠
انځور ٧- ٣- ٢٠١
انځور ٧- ۴- ٢٠٢

انځور ۷- ۵- ۲۰۳
انځور ۷- ٦- ۲۰۴
انځور ۷- ۷- ۲۰۷
انځور ۷- ۸- ۲۰۷
انځور ۷- ۹- ۲۰۹
انځور ۷- ۱۰- ۲۱۰
انځور ۷- ۱۱- ۲۱۴
انځور ۷- ۱۲- ۲۱۴
انځور ۷- ۱۳- ۲۱۵
انځور ۷- ۱۴- ۲۱٦
انځور ۷- ۱۵- ۲۱۷
انځور ۷- ۱٦- ۲۱۷
انځور ۷- ۱۷- ۲۱۹
انځور ۷- ۱۸- ۲۱۹
انځور ۷- ۱۹- ۲۲۱
انځور ۷- ۲۰- ۲۲۲
انځور ۷- ۲۱- ۲۲۲
انځور ۷- ۲۲- ۲۲۴
انځور ۷- ۲۳- ۲۲۴
انځور ۷- ۲۴- ۲۲٦
انځور ۷- ۲۵- ۲۲۷
انځور ۸- ۱- ۲۳۸
انځور ۸- ۲- ۲۴۲
انځور ۸- ۳- ۲۴٦

انځور ۸- ۴- ۲۴۶
انځور ۸- ۵- ۲۵۴
انځور ۸- ٦- ۲۵۴
انځور ۸- ۷- ۲۵۵
انځور ۸- ۸- ۲۵۵
انځور ۸- ۹- ۲۵۵
انځور ۸- ۱۰- ۲۵۷
انځور ۸- ۱۱- ۲۵۸
انځور ۸- ۱۲- ۲۵۸
انځور ۸- ۱۳- ۲٦۲
انځور ۸- ۱۴- ۲٦۲
انځور ۸- ۱۵- ۲٦۳
انځور ۹- ۱- ۲۸۳
انځور ۹- ۲- ۲۸۴
انځور ۹- ۳- ۲۸۴
انځور ۹- ۴- ۲۸۴
انځور ۱۰- ۱- ۲۹۲
انځور ۱۰- ۲- ۲۹۲
انځور ۱۰- ۳- ۲۹۳
انځور ۱۰- ۴- ۲۹۳
انځور ۱۰- ۵- ۲۹۵
انځور ۱۰- ٦- ۲۹٦
انځور ۱۰- ۷- ۲۹٦
انځور ۱۰- ۸- ۲۹۷

انځور ۱۰- ۹- ۲۹۷
انځور ۱۰- ۱۰- ۲۹۸
انځور ۱۰- ۱۱- ۲۹۸
انځور ۱۰- ۱۲- ۲۹۹
انځور ۱۰- ۱۳- ۲۹۹
انځور ۱۰- ۱۴- ۳۰۱
انځور ۱۰- ۱۵- ۳۰۲
انځور ۱۰- ۱٦- ۳۰۳

د جدولونو لیکلړ

جدول ۱- ۱- ٦
جدول ۱- ۲- ۱۲
جدول ۱- ۳- ۱۳
جدول ۱- ۴- ۲۳
جدول ۱- ۵- ۲۵
جدول ۱- ٦- ۲۷
جدول ۱- ۷- ۲۸
جدول ۲- ۱- ۴۴
جدول ۲- ۲- ۴۴
جدول ۲- ۳- ۴۵
جدول ۲- ۴- ۴٦
جدول ۲- ۵- ۴۷
جدول ۲- ٦- ۴۷
جدول ۲- ۷- ۴۸
جدول ۲- ۸- ۴۸
جدول ۲- ۹- ۵۵
جدول ۲- ۱۰- ۵٦
جدول ۲- ۱۱- ۵۸
جدول ۳- ۱- ٦۸
جدول ۳- ۲- ٦۹
جدول ۳- ۳- ٦۹
جدول ۳- ۴- ۷۰

جدول ۳- ۵- ۷۳
جدول ۳- ٦- ۷۳
جدول ۳- ۷- ۷۴
جدول ۳- ۸- ۷۵
جدول ۴- ۱- ۱۰۱
جدول ۴- ۲- ۱۰۲
جدول ۴- ۳- ۱۰۳
جدول ۴- ۴- ۱۰۵
جدول ۴- ۵- ۱۰۸
جدول ۴- ٦- ۱۱۱
جدول ۴- ۷- ۱۱۲
جدول ۴- ۸- ۱۱۷
جدول ۴- ۹- ۱۱۷
جدول ۴- ۱۰- ۱۱۹
جدول ۴- ۱۱- ۱۲۰
جدول ۴- ۱۲- ۱۲۵
جدول ۴- ۱۳- ۱۲۷
جدول ۵- ۱- ۱۳۸
جدول ۵- ۲- ۱۴۰
جدول ۵- ۳- ۱۴۰
جدول ۵- ۴- ۱۴۲
جدول ۵- ۵- ۱۴٦
جدول ۵- ٦- ۱۴۸
جدول ۵- ۷- ۱۴۹

جدول ۵- ۸- ۱۵۱
جدول ۵- ۹- ۱۵۲
جدول ۵- ۱۰- ۱۵۲
جدول ۵- ۱۱- ۱۵۴
جدول ۵- ۱۲- ۱۵۴
جدول ۵- ۱۳- ۱۴۵
جدول ٦- ۱- ۱۷۵
جدول ٦- ۲- ۱۷۵
جدول ٦- ۳- ۱۸۲
جدول ٦- ۴- ۱۸۴
جدول ٦- ۵- ۱۸۷
جدول ٦- ٦- ۱۹۱
جدول ۷- ۱- ۲۰۱
جدول ۷- ۲- ۲۰۴
جدول ۷- ۳- ۲۰۹
جدول ۷- ۴- ۲۱۰
جدول ۷- ۵- ۲۱۰
جدول ۱۰- ۱۷- ۳۰۴
جدول ۱۰- ۱۸- ۳۰۴
جدول ۱۰- ۱۹- ۳۰۵

لومړی څپرکی

سیگنلونه او د اعدادو سیستمونه

(Signals And Number Systems)

موخې: د دغه څپرکي په پای کې به تاسو وکولی شئ چې:

- د یو کمپیوټربنسټیزې برخې تشریح کړئ.
- د انالوگ او ډیجیټل سیگنل تر منځ توپیر وکړئ.
- د یو سیگنل ځانگړتیاوې زده کړئ .
- د لســو پـه قـاعـده د اعـدادو بدلول، د دوه قاعدې لرونکو اعدادو ته او بېرته د دې برعکس.
- د دوو په قاعدې د اعدادو جمع او منفي کول .
- د دوو په قاعدې اعدادو کې د اعشاري اعدادو ښودل.
- د دوو په قاعدې اعدادو بدلول د شپاړلسو په قاعدې اعدادو ته.
- د موازي او مسلسل جریان توپیر.

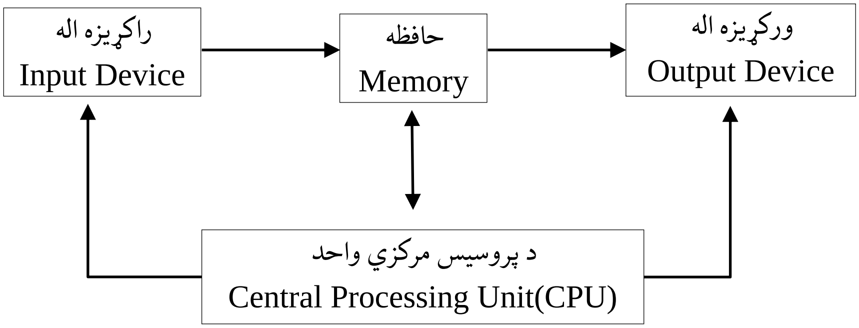
۱.۱ پېژندنه

اعـداد زموږ د ورځني ژونـد یو نه بېلېدونکې برخه ده. اعداد په انالوګ یا ډیجیټل طریقه باندې ښودل کېږي؛ د مثال په ډول د انالوګ ساعت، ډیجیټل ساعت او یا ترمامیټر.

د ډیجیټل سیګنل ګټې نسبت انالوګ سیګنل ته:

۱. د ډیجیټل اعداد ډېر دقیق وي.
۲. د ډیجیټل معلوماتو ذخیره کول اسانه وي.
۳. د ډیجیټل سیستمونو ډیزاین کول اسانه وي.
۴. غیر ضروري څپې (Noise) ورباندې لږ تاثیر لري.
۵. ډیجیټل سیستمونه په اسانۍ سره یو مدغم شوي سرکټ (Circuit Integrated) کې پروګرام کېږي.

ډیجیټل سیګنل په وقفه یي شکل سره وي او انالوګ سیګنل په متواتر شکل سره د معلوماتو لېږد، رالېږد ترسـره کوي. ډیجیټل سـیسـتمونه په پراخه پیمانه اسـتعمالېږي چې زیاتره پرواګرامونه یې په کمپیوټر، حسـاب ماشـین او موبایل تیلیفون کې تر سترګو کېږي. په ډیجیټل سیستمونو کې د معلوماتو جریان د ډیجیټل سیستمونو د مختلفو برخو تر منځ د ډیجیټل سیګنل (Digital Signal) په شکل لېږل کېږي.



۱.۱ انځور: د کمپیوټر اساسي برخې

کمپیوټر له دوو برخو څخه جوړ شوی دی: سختغالو (Hardware)، پوستغالو (Software). سختغالی (Hardware) د کمپیوټر فزیکي برخې بیانوی لکه کیبورډ، د پروسیس مرکزي واحد (CPU) او حافظه. پوستغالې (Software) هغه پروګرام ته وایي چې د پروسیس مرکزي واحد (CPU) له خوا چلول کېږي لکه د کمپیوټر عامل سیستم (Operating Systems) او کاریال پروګرامونه (Application Programs). کمپیوټر په مختلفو شکلونو کې شتون لري لکه: ډیسکټاپ، لپ ټاپ، ټیبلیټ، سرور او ایفون. د مختلفو شکلونو په پام کې نیولو سره ټول کمپیوټرونه یو بل ته ورته اساسي جوړښت لري. انځور: ۱.۱ د کمپیوټر اساسي برخې څرګندوي.

ورکړیزه اله (Input Device): ورکړیزه اله حافظې ته د معلوماتو ورکولو له پاره استعمالېږي. د ورکړیزې الې (Input Device) مثالونه لکه کیبورډ، موس، د ټچ پېډ سکرین، سپک قلم (Light Pen)، بارکوډ لوستونکی او سکنر شامل دي. ورکړیزه اله (Input Device) معلومات په بیټ (Bits) بدیلوي او بیټونه (Bits) په حافظه کې زېرمه کېږي.

راکړیزه اله (Output Device): د کمپیوټر حافظه معلومات د بیټ (Bit) په شکل راکړیزې الې ته لیږي، راکړیزې الې دا بیټونه په تورو (Characters)، انځورونو او غږونوباندې تبدیلوي، ترڅو انسانان پرې پوه شي.

حافظه: حافظه د دې له پاره استعمالېږي چې معلومات او پروګرامونه ذخیره کړي. د کمپیوټر په حافظه جامد شکل وي لکه ربم (RAM)، یوازې لوستونکې زېرمه (ROM)، فلش یا هارډسک.

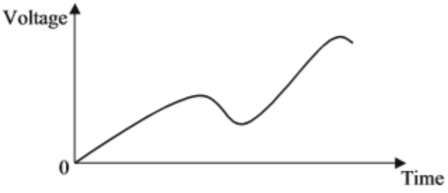
پروسیس مرکزي واحد (CPU): د پروسیس مرکزي واحد د دې له پاره استعمالېږي چې په معلوماتو محاسبوي عملیې اجرا کړي.

۲.۱ انالوګ سېګنل (Analog Signal)

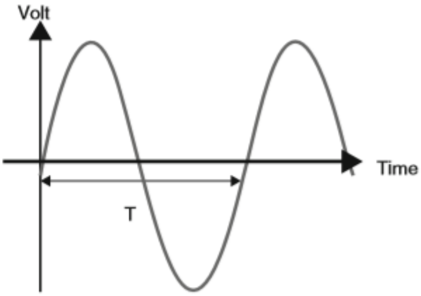
انالوګ سـېګنل هغه سـیګنل دی چې اندازه یې د وخت فعالیت دی او د وخت په بـدلیـدو سـره ورو ورو بدلیږي. انالوګ سـېګنل په مهالي (Periodic) او غېرې مهالي (Non Periodic) سېګنل باندې وېشل شوي دي.

غیرې مهالي سـېګنـل (Nonperiodic Signal): په غیرې مهالي سـیګنل کې تکراري نمونې شتون نه لري لکه ۲.۱ انځور: کې چې ښودل شوي.

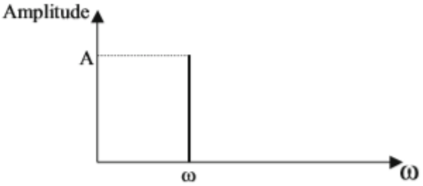
مهالي سـېګنل (Periodic Signal): هغه سـېګنل دی چې په یو معلوم وخت کې تکراري نمونې پـه کې رامنځتـه کیږي او د بشـپړې نمونې تکرارتـه دوره (*Cycle*) ویـل کېږي. تر ټولو سـاده دوراني سـېګنل جاري څپه (Wave Sine) ده، هغه چې په ۳.۱ انځور: کې ښـودل شـوې. د وخت په محدوده (Domain) کې څپې لمن د طول اندازه a(t) ده، چې په ریاضـی کې $a(t) = Sin(\omega P \theta)$ باندی ښـودل کېږي. او A د اعظمي امپلیتود (Amplitude)، ω زاویـه وي فریکونســي ده او θ د پړاو (Phase) زاویه ده.



۲.۱ انځور: دا د غیر ې مهالي انالوګ سېګنال ښيايي.



۱.۳ انځور: د وخت حدود د سیګنال څپې.



۴.۱ انځور: د فریکونسي د زوړ څپې ښايي

مهالي سيګنل فريکونسي ډومين (Domain) کې هم ښودل کيږي، چې افقي محور فريکونسي ده او عمودي محور د سيګنل امپليتود دی. ۴.۱ انځور د يو جاري سيګنل (Sine Signal) فريکونسي ډومين ښايي.

يو بريښنايي سيګنل چې معمولاً غږ، تودوخه يا ميوزيکي غږ ښايي چې له څو څپو څخه جوړ شوی وي. دا سيګنلونه يو اساسي فريکونسي او څو فريکونسي لري چې د اواز پوهنې (Harmonics) په نوم يادېږي.

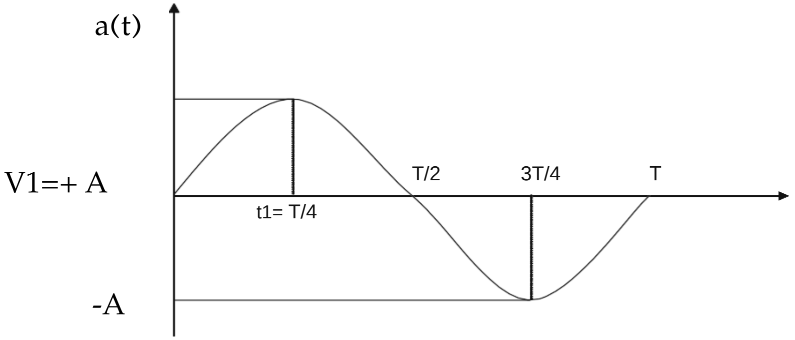
۱.۲.۱ د انالوګ سيګنل ځانګړنې

(Characteristics Of An Analog Signal): د مهالي انالوګ سيګنل ځانګړتياوې عبارت دي له فريکونسي، امپليتود او پړاو (Phase) څخه.

فریکونسي: فریکونسي (F) د دورانونو (Cycles) تعداد په 1 S، F = 1/T چې د Hz(Hertz) په واسطه ښودل کېږي. که چېرې د انالوګ سېګنل هر دوران په 1s کې تکرار شي، نو د سېګنل فریکونسي یې 1Hz ده. که انالوګ سېګنال (۱۰۰۰) ځله په ثانیه (یو ځل په هر میلي سیکېنډ) کې تکرارشي. فریکونسي به:

Units Of Frequency	Numerical Value	Units Of Period	Numerical Value
Hertz (Hz)	1 Hz	Second (S)	1 S
Kilohertz (Khz)	10^3 Hz	Millisecond (Ms)	10^{-3} S
Megahertz (Mhz)	10^6 Hz	Microsecond (Ms)	10^{-6} S
Gigahertz (Ghz)	10^9 Hz	Nanosecond (Ns)	10^{-9} S
Terahertz (Thz)	10^{12} Hz	Picosecond (*P*s)	10^{-12} S

۱.۱ جدول: فریکونسي او دوران ځانګړي واحدونه

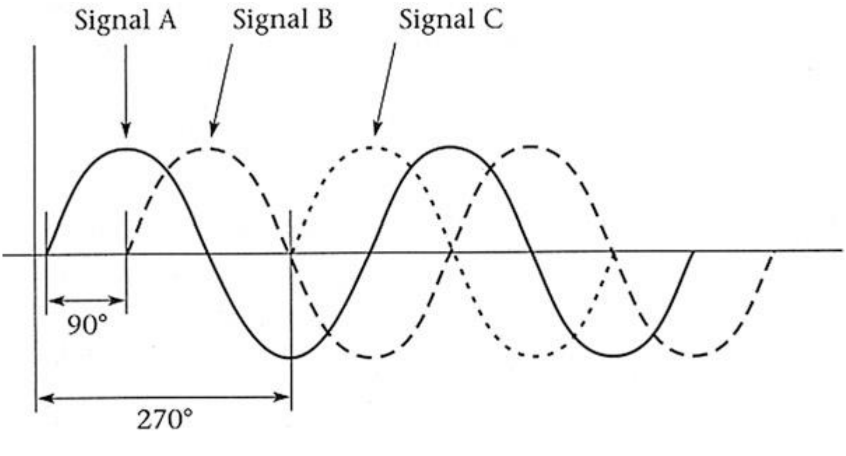


۵.۱ انځور: په یو دوران کې جاري څپه (Sine Wave)

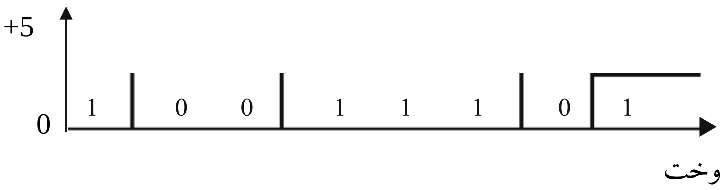
د اهتزاز لمن (Amplitude): د انالوگ سـیگنال د طول موده، د وخت په فعالیت اندازه کېږي، لکه څنگه چې په ۵.۱ انځور: کې ښودل شوې، شاید په ولټ (د ولتیج واحد) سره هم و ښودل شي. په بل عبارت د اهتزاز لمن د هغه د ولتیج اندازه په یوه خاص وخت کې ده. وخت ده t1 او د سېگنل د اهتزاز لمن V1 ده.

پړاو (Phase): دوه سېگنلونه په عینې فریکونسي سره کېدای شی مختلف پړاونه ولري. د دی معنی دا ده چې یو له دې سېگنلونو څخه د بل سېگنل څخه په مختلف وخت کې شروع کېږي. دا فرق (0° To 360°) درجو او رادیانونو په واسطه ښودل کېږي. د پړاو 0° درجه دا څرگندوي چې جاري څپه په صفر وخت کې شروع شوې ده او د پړاو 90 درجې دا ښایي چې سېگنل په 90° شروع کېږي لکه څنگه چې په انځور: ۱.۶ کې ښودل شوی.

۱.۱ مثال: د جاري څپې (Sine Wave) له پاره معادله پیدا کړئ، که چېرې فریکونسي 10Hz وي اعظمي د اهتزاز لمن یې 20 volts وي او فاز یې 0° وي.



۱.٦ انځور: درې ساین څپې په مختلف فاز کې



۷.۱ .انځور: ډیجیټل سېګنل

$$\omega = 2\pi f = 2 \times 3.1416 \times 10 = 62.83 \frac{rad}{\sec}$$
$$a(t) = 20 \sin(62.83\, t)$$

۳.۱ ډیجیټل سېګنلونه (Digital Signals)

پرمختللي کمپیوټرونه ارتباطات په ډیجیټل سېګنلونو تامینوي. ډیجیټل سېګنلونه د دوو ولتېجونو په واسطه ښودل کېږي، یو ولتېج په 0 نمبر ښودل کېږي د دوو په قاعدې اعداد (Binary Number) کې او بل ولتیج د 1 په واسطه ښودل کېږي. د ډیجیټل سېګنل یو مثال په ۷.۱ انځور: کې ښودل شوی، چېرته چې د دوو په قاعدې کې 0 ولتېج ښودل کیږي په 0 سره او 5+ ولټونه 0.1 یا 1 کوم ته چې بیټ ویل کېږي او 8 بیټو ته یو بایټ (Byte) وایي.

۴.۱ د اعدادو سیستم (Number System)

اعداد په مختلفو قاعدو ښودل کېږي. د لسو قاعدې ته ډیسیمل (Decimal) وایي په لاندې مثال کې (356) په ډیسیمل ښودل شوي.

$$356 = 3 * 10^2 + 5 * 10^1 + 6 * 10^0$$

په عمومي توګه کولای شو نمبرونه په لاندې انځور: سره وښایو،

$$(a_5 a_4 a_3 a_2 a_1 a_0 . a_{-1} a_{-2} a_{-3})_r,$$

چې r د اعدادو قاعده ده او a_i بايد کم وي له r څخه.

$(10011)_2$ دا يو صحيح نمبر دی او $(211.01)_2$ صحيح نمبر نه دی.

معادله ۱.۱ د اعدادو د هرې راکړل شوې قاعدې څخه د لسو قاعدې ته د تبديلولو له پاره استعمالېږي.

$$\underbrace{(a_5a_4a_3a_2a_1a_0}_{\text{Integer}}.\underbrace{a_{-1}a_{-2}a_{-3})_r}_{\text{Fraction}} = a_0 \times r^0 + a_1 \times r^1 + a_2 \times r^2 + a_3 \times r^3 + \ldots + a_{-1} \times r^{-1} + a_{-2} \times r^{-2} + \cdot a_{-2} \times r^{-3} \ldots \quad (1.1)$$

۲.۱ مثال: $(27.35)_8$ د لسو قاعدې اعدادو ته واړوئ:

$$(27.35)_8 = 7*8^0 + 2*8^1 + 3*8^{-1} + 5*8^{-2} = 7 + 16 + .375 + .078125 = (23.45)_{10}$$

۳.۱ مثال: 1101111 د لسو قاعدې (Decimal) ته واړوئ:

$$(1101111)_2 = 1*2^0 + 1*2^1 + 1*2^2 + 1*2^3 + 0*2^4 + 1*2^5 + 1*2^6 = 1 + 2 + 4 + 8 + 32 + 64 = (111)_{10}$$

۱.۴.۱ د دوو له قاعدې څخه د لسو قاعدې ته تبديلول (Converting From Binary To Decimal):

معادله ۱.۲ د دوو په قاعدې د اعدادو (*Binary Number*) عمومي شکل ښايي.

$$(a_5a_4a_3a_2a_1a_0.a_{-1}a_{-2}a_{-3})_2 \quad (1.2)$$

ai د دوه قاعدې يو عدد (Binary Digit) او يا بيټ (Bit) دی.

$$\underbrace{(a_5a_4a_3a_2a_1a_0}_{\text{Integer}}.\underbrace{a_{-1}a_{-2}a_{-3})_r}_{\text{Fraction}} = a_0 \times r^0 + a_1 \times r^1 + a_2 \times r^2 + a_3 \times r^3 + \ldots + a_{-1} \times r^{-1} + a_{-2} \times r^{-2} + \cdot a_{-2} \times r^{-3} \ldots \quad (1.1)$$

۲.۱ معادله چې د لسو قاعدې ته يي د ۱.۱ معادلې په واسطه تبدیل کړو.

$$(a_5a_4a_3a_2a_1a_0.a_{-1}a_{-2}a_{-3})_2 = a_0 + 2\,a_1 + 4\,a_2 + 8\,a_3 + 16\,a_4 + 32\,a_5 + 64\,a_6 + \frac{1}{2}*a_{-1} + \frac{1}{4}*a_{-2} + \frac{1}{8}*a_{-3}$$

۱.۴ مثال: د $(110111.101)_2$ تبدیلول د لسو په قاعدې اعدادو ته.

$$(110111.101)_2 = 1*2^0 + 1*2^1 + 1*2^2 + 0*2^3 + 1*2^4 + 1*2^{*5} + 1*2^{-1} + 0*2^{-2} + 1*2^{-3} = 55.625$$

یا

2^5	2^4	2^3	2^2	2^1	2^0		2^{-1}	2^{-2}	2^{-3}
1	1	0	1	1	1	.	1	0	1

کـه چېرې د $32 + 16 + 0 + 4 + 2 + 1 + 1/2 + 0 + 1/8$ دوو د قـاعـدې اعداد د N بیټونو د یو جوړ وي نو بیا يې د لسو د قاعدې اعداد يې 2^{n-1} وي.

۵.۱ مثال:

$11 = 2^2 - 1 = 3$
$111 = 2^3 - 1 = 7$
$1111 = 2^4 - 1 = 15$
$11111 = 2^5 - 1 = 31$
$111111 = 2^6 - 1 = 63$

دوه ییز یـا د دوو په قاعدې اعداد په *0* او *1* ســره ښــودل کېږي. د دوه په قاعدې اعداد 0 او 1 چې بیټونه (Bits) ورته ویل کېږي، 8 بیټونه (Bits) ته یو بایټ (Byte) ویـل کېږي، 16 (Bits) بیټونو تـه نیم لغـت یـا ویي ویل کېږي او 4 بایټونو (Byte) ته پوره لغت ویل کېږي.

۲.۴.۱ د تامو اعدادو بدلون د لسو قاعدې څخه د دوه په قاعدې اعدادو ته (Converting From Decimal Integer To Binary): د دې له پاره چې تام اعداد د لسو قاعدې (Decimal) څخه د دوو قاعدې (Binary) ته بدل کړو، نو د لسو قاعدې اعداد په غوښتل شوې نوې قاعدې باندې وېشوو (د دوو قاعدې اعدادو باندې). په نتیجه کې به یا پوره تقسیمېږي یا به یو باقې پاتې کېږي (0او یا 1 وي). لومړی پاتې قیمت تر ټولو لږ اهمیت لرونکی قیمت دی، همداسې لاسته راغلي اعداد په مسلسل ډول تقسیموو تر څو چې باقي عدد صفر 0 لاسته راشي.

۱.٦ **مثال:** د لسو قاعدې 34 عدد بدلون د دوه قاعدې ته:

	Quotient	Remainder
34/2 =	17	$0 = a_0$
17/2=	8	$1 = a_1$
8/2	4	$0 = a_2$
4/2	2	0 = a3
2/2	1	$0 = a_4$
1/2	0	$1 = a_5$
Therefore $34 = (100010)_2$		

که چېري د تقیسم حاصل ټول یو وه نو بیا $2^n - 1$ معادلې په استعمال سره دا د لسو قاعدې ته بدلیدلای شي.

مثالونه:

Binary number	$2^n - 1$	Decimal number
11	2^2-1	3
111	2^3-1	7
1111	2^4-1	15
11111	2^5-1	32

د دوو قاعدې اعداد په $a_5\ a_4\ a_3\ a_2\ a_1\ a_0$ سره ښودل کیږي چې a_0 بیا 2^0 وي ، a_1 بیا 2^1 وي او a_5 بیا 2^5 وي. جدول ۲.۱ کې 2^n ښودل شوی.

۳.۴.۱ د لسو قاعدې د کسري (اعشاري) اعدادو تبدیلول د دوو قاعدې ته (Converting Decimal Fraction To Binary):

د لسو قاعدې اعداد په $(0.XY)_{10}$ سره ښودل کېږي کولای شو دا د دوو قاعدې ته تبدیل کړو چې بیا په $(0.a_{-1}, a_{-2}, a_{-3}, etc.)_2$ سره ښودل کېږي.

کسري اعداد په ۲ کې ضربیږی، چې د ثابتې برخې نتیجه یې a_{-1} ده او کسري برخه یې بیا په ۲ کې ضربیږي، چې د ثابتې برخې نتیجه یې a_{-2} ده او کسري برخه یې بیا په دوو کې ضربوو په همدې ترتیب تر هغې ادامه ورکوو تر څو کسري برخه صفر شي.

$(?)_2 = (0.35)_{10}$

0.35*2	=	0.7	=	0	+	0.7	$a_{-1} = 0$
0.7*2	=	1.4	=	1	+	0.4	$a_{-2} = 1$
0.4*2	=	0.8	=	0	+	0.8	$a_{-3} = 0$
0.8*2	=	1.6	=	1	+	0.6	$a_{-4} = 1$
0.6*2	=	1.2	=	1	+	0.2	$a_{-5} = 1$

کله، کله کسري (اعشاري) برخه نه صفر کېږي او د بیټونو (Bits) تعداد چې د کسر له پاره استعمالیږي د هغه په دقت باندې دلالت کوي، چې د استعمالونکي له خوا معرفي شوي دي، نو څکه 0.35 د دوو په قاعدې کې 0.010011 کېږي

جدول ۱.۲ : 2^n د N د مختلفو قیمتونو سره

د لسو قاعدې عددونه	2^n	د لسو قاعدې عددونه	2^n	د لسو قاعده	2^n
64 K=65,536	2^{16}	256	2^8	1	2^0
128 K=131,072	2^{17}	512	2^9	2	2^1
256 K=262,144	2^{18}	1 K=1024	2^{10}	4	2^2
512 K=524,288	2^{19}	2 K=2048	2^{11}	8	2^3
1 M=1,048,576	2^{20}	4 K=4096	2^{12}	16	2^4

2^5	32	2^{13}	8 K=8192	2^{21}	2 M
2^6	64	2^{14}	16 K=16,384	2^{22}	4 M
2^7	128	2^{15}	32 K=32,768	2^{23}	8 M

جدول ۱.۳: د لسو قاعدې اعداد د دوو قاعدې او شپاړسو قاعدې معادل

د لسو قاعده	د دوو قاعده (Base Of 2)	د شپاړسو قاعده (Base Of 16) Or HEX
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

د شپاړسو په قاعدې اعدادو سیستم 16 سمبولونه لري (د 0 نه تر 9 پورې او بیا د A نه تر F پورې). جدول ۳.۱ د لسو قاعدې عددونه، د دوو قاعدې عددونه د 0 نه تر 15 پورې او د شپاړسو قاعدې اعدادو معادل ښايي.

۴.۴.۱ د شپاړسو قاعدې اعدادو تبدیلول د دوو قاعدې اعدادو ته (Converting From Hex to Binary):

کولای شو چې ۳.۱ جدول د شپاړسو قاعدې څخه د دوه قاعدې ته اعداد تبدیلولو له پاره استعمال کړو او هم د دوه له قاعدې څخه د شپاړسو قاعدې ته اعداد تبدیلولو له پاره.

مثال ۷.۱ : د دوو قاعدې عدد 001010011010 تبدیلول د شپاړسو قاعدې اعدادو ته. د دې له پاره له ښي څخه چپ طرف ته څلور، څلور بیټونه (Bits) ګروپ کوو. د ۲.۲ جدول په استعمال سره، نو د دې له پاره د هر څلور بیټونو (Bits) معادل شپاړس په قاعدې عدد لیکو.

مثال ۸.۱: د $(3D5)_{16}$ بدلول د دوو قاعدې ته. د ۲.۲ جدول په استعمال سره چې نتیجه به یې د دوه په قاعدې اعدادو کې عبارت وي له:

3	D	5
0011	**1101**	**0101**

چې حاصل یې عبارت دی له 001111010101 څخه.

0010	1001	1010
2	**9**	**A**

۱۰.۱ مثال: د 6DB بدلول د شپاړسو له قاعدې څخه د دوو قاعدې ته د ۳.۱ جدول په استعمال سره، چې نتیجه به یې د دوو په قاعدې اعدادو کې عبارت وي له:

6	D	B
0110	1101	1011

چې حاصل یې عبارت دی له 011011011011 څخه.

۱. ۱۱ مثال: د $(110111.101)_2$ بدلول د لسو قاعدې ته

$$(110111.101)_2 = 1*2^0 + 1*2^1 + 1*2^2 + 0*2^3 + 1*2^4 + 1*2^{*5} + 1*2^{-1} + 0*2^{-2} + 1*2^{-3} = 55.625$$

۵.۴.۱ د دوه په قاعدې اعدادو جمع کول(Binary Addition)

$$1+0=1\ ,\ 1+1=10$$

$$\begin{array}{lr} \textit{Carry bits} & 111 \\ & 10101 \\ & +\ 01101 \\ \hline & 100010 \end{array}$$

د دوو په قاعدې اعدادو کې له چپ طرف څخه لومړنی بیټ (Bit)، ته ډېر مهم بیټ (MSb) وايي. او له ښي طرف اعدادو څخه لومړنی بیټ (Bit) ته لږ مهم بیټ (LSb) وايي.

$$\underline{\text{MSb}} \rightarrow 1\ 0\ 0\ 1\ 0 \leftarrow \underline{\text{LSb}}$$

۵.۱ کمپلیمینټ (Complement) او دوه ییز کمپلیمینټ (Two's Complement)

د 1 کمپلیمینټ 0 دی او د 0 کمپلیمنټ 1 دی.

د دوه په قاعدې اعدادو د کمپلیمنټ د پیدا کولو له پاره د هر بیټ (bit) کمپلیمنټ نیسو.

۱۲.۱ مثال: د 101101 کمپليمنټ 010010 دی.
د يو عدد دوه ييز کمپلمينټ = د يو عدد کمپليمينټ + 1 سره

۱۳.۱ مثال: دوه ييز کمپليمنټ د 101011 عدد مساوي دی له:

010100 (کمپليمنټ) + 1 =010101

۱۴.۱ مثال: د 10000 دوه ييز کمپليمنټ پيدا کړئ؟

01111 (کمپليمنټ) + 1 =10000

۱.۵.۱ د دوه غيرې علامه لرونکو عـددونو تفريق کول د دوه ييز کمپليمينـټ پـه طـريقـه(Subtraction of Unsigned Number Using Two's Complement): د تفريق له پاره يې لاندې کړنلاره استعماليږي

$A = a_5\ a_4\ a_3\ a_2\ a_1\ a_0$ څخه $B = b_5\ b_4\ b_3\ b_2\ b_1\ b_0$:

۱. د B دوه ييز کمپليمنټ جمع کول A ته.
۲. وګورئ که چېرته په نتيجه کې حاصل پاتی شو:

(a) که چيري په نتيجه کې حاصل پاتې شو، نو حاصل ايسته کړئ او نتيجه يې مثبت ده.

(b) که چېرې په نتيجه کې حاصل پاتې نه شو، نو د نتيجې دوه ييز کمپليمنټ واخلئ او نتيجه منفي ده.

۱۴.۱ مثال: A=110101 څخه B=101010 منفي کړئ؟
لومړی B کمپليمنټ نيسو010101
اوس يې د 1 سره جمع کوو:

Two's Complement Of B=010101+1=010110

Add Two’s Complement Of B To A

```
   110101
+  010110
----------------
   10 01011
```

Carry, discard the carry and the result is +001011.

۱۵.۱ مثال: د A=101010 څخه B=110101 تفریق کړئ:

د B دوه ییز کمپلیمنټ 001010+1=001011.

دوه ییز کمپلیمنټ د B له A سره جمع کړئ.

```
  001011
+ 101010
--------
  110101
```

لکه څرنګه چې موږ وینو، که چېرته دوه شــپږ 6 بیټ (Bit) عددونه په خپل منځ کې جمع کړو نو نتیجه یې هم یو 6 بیټ (Bit) عدد په لاس راځي. دلته هېڅ حاصل نشته، نو موږ فقط د نتیجې دوه ییز کمپلیمنټ نیسو.

Two’s Complement Of 110101 = 001010 + 1 = - 001011

دوه ییز کمپلیمنټ د 110101=001010+1= - 001011

۶.۱ بې علامې ، علامه لرونکي، ميګنيچيوډ، او د دوه په قاعده دوه ييز کمپليمنټ عدد (Unsigned, Signed Magnitude, and (Signed Two’s Complement Binary Numbe

کولای شــو چې د دوه پــه قاعده عددونه د بې علامې يا علامه لرونکو او يا هم د دوه ييز کمپليمنټ عددونو په شـکل وښـايو + علامه په 0 سره او – علامه په 1 سره ښودل کيږي.

د بې علامې عدد (Unsigned Number) ټول بيټونه (Bits) يو عدد ښايي مګر علامه لرونکي عددنو، مهم (Significant) بيټ (Bit) د يو عدد علامه ښــايي. 1 چې کله د اعدادو په مهم ځای کې راشــي نو د منفي علامی معنی ورکوي او 0 چې کله د اعدادو په مهم ځای کې راشي د مثبت علامې معنی ورکوي.

د 1101 بې علامې (Unsigned) قيمت 13 دی.

علامـه لرونکي مګنيچيوډ عـدد پـه علامه لرونکي عدد کې يو مهم (Significant) بيټ (Bit) د عدد علامه ښايي، چې

1101= - 5 Or 0101 =5 +5 علامه لرونکي عدد کې 1101=13 په بې علامې عدد کې.

علامه لرونکی (Signed) دوه ييز کمپليمنټ، علامه لرونکی دوه ييز کمپليمنټ په منفي عـدد عملي کيږي. کـه چيري د يو عـدد علامه يو وه، نو بيا عدد په علامه لرونکي دوه ييز کمپليمنټ سره ښودل کيږي.

۱٦.۱ مثـال: 10 (5-) پـه علامـه لرونکي دوه ييز کمپليمنـټ (Complement) کې د څلورو بيټونو (Bits) په واسطه ښودل کيږي.

10(5-) پـه علامـه لرونکي عـدد کې 1101 دوه ييز کمپليمنټ د 101 برابر دی له 011 سـره او د علامې د بيټ (Bit) په اضـافه کولو سره به نتيجه 1011 وي، چې ښودل کيږي 5- په علامه لرونکي دوه ييز کمپليمنټ کې.

۱۷.۱ مثال: 10(23-) د 8 بیټ (Bit) د دوه ییز کمپلیمنټ سره ښودل کېږي.

$(23)_{10}$ = $(1\ 0\ 1\ 1\ 1)_2$ په بې علامه د دوو په قاعدې.

۷.۱ د دوو په قاعدې د اعدادو جمع کول د علامه لرونکي دوه ییز کمپلیمنټ په ذریعه (Binary Addition Using Signed Two's Complement)

لاندې مثال د دوو علامه لرونکو عددونو د جمعې نتیجه ښايي.

$$(4) + (+3) + (A)$$

دواړه عددونه د دوو قاعدې ته واړوئ، مهم بیټ (Bit) یې علامه ښايي او نتیجه یې مثبت ده.

$$0011 + 0100 = 0111 = +7$$

$(b)\ (+3)10 + (-4)10$

بیا $(-4)_{10}$ په علامه لرونکي دوه ییز کمپلیمنټ کې ,1100 داسې وي.

$0011 + 1100 = 1111;$

نتیجه کې حاصل لاس ته نه راځي باید دوه ییز کمپلیمنټ واخلو چې نتیجه 1- وي

$(-3) + (+5) = 1101 + 0101 = 1\ 0010$

(حاصل لرې کړئ او نتیجه یې 2+)

(C) $(-7)_{10} + (-5)_{10}$

په 4 بیټ (Bit) علامه لرونکي دوه ییز کمپلیمنټ کې د دواړو عددونو ښودل:

$(-7)_{10} = 1001$

$(-5)_{10} = 1011$

1001 + 1011 = 10100

↑ Sign Bit

د دوه مثبت علامه لرونکو عددونو د جمع کولو په صورت کې حاصل مثبت لاسته راځي چې دا عمل د توېېدو (Overflow) په نوم يادېږي.

$$(+7) + (+6) = (0111) + (0110) = 1101$$

د دوه مثبت علامه لرونکو عددونو د منفي کولو په صورت کې حاصل منفي لاسته راځي چې دا عمل د توېېدو (Overflow) په نوم يادېږي.

۱۸.۱ مثال: لاندې جمع کول د 8 بيټ (Bit) علامه لرونکي دوه ييز کمپليمنټ په ذريعه. د جمعي توېېدنې (Overflow) حالتونه : په لاندې حالتونو کې د دوه علامه لرونکو عددونو نتيجه توېېدنه (Overflow) ده:

$(-38)_{10} + (44)_{10}$

$(-38)_{10}$ in signed two's complement = 11011010
$(+44)_{10}$ + 00101100
100000110 discard the carry and result is 00000110 = 6

B. Add −38 to −44 using 8 bit signed two's complement
−38 = 11011010
−44 = 11010100

10101110 = −82

C. Add +100 to +44
100 = 01100100
44 = 00101100

10010000 the sign of result is negative then results produce overflow

(a) که چېرې دواړه عددونه منفي وي او د جمعې نتيجه يې مثبته په لاس راشي.

$$(-A) + (-B) = +C$$

(b) که چېرې دواړه عددونه مثبت وي او د تفريق نتيجه يې منفي په لاس راشي.

$$(+A) + (+B) = -C$$

۸.۱ د اعشاري اعدادو ښودل (Floating Point Representation)

د کمپيوټر د پروسيس مرکزي واحد (CPU) عموماً د منطقي حسابي برخې (ALU)، اعشارې برخه (FLU/FPU)، رجسټر (Register)، کنټرول برخه او د کېشې (Cache) حافظې برخې لرونکې ده.

د منطقي حسابي برخه (ALU) د تام اعدادو حسابي عمليې اجرا کوي لکه جمع، منفي او منطقي عمليې لکه اينډ (AND)، اور (OR)، ايکس ار (XOR) او داسې نور. تام (Integer) عدد يو مکمل عدد بې له کسري يا اعشاري برخې لکه ۱، ۲ او ۳ په داسې حال کې ۱.۰، ۲.۲، او 3.0001 دا ټول کسري اجزا لري چې اعشاري (Floating Point) اعداد ورته وايي.

اعشاري برخه (Floating Point Unit) د اعشارې عمليې ترسره کوي، اعشاري اعداد د علامې، مانتيزا (Mantissa) او اکسپونينشل برخې درلودونکې دي. د برقي او د برق د انجينرانو انستيتوت (IEEE) د اعشاري اعدادو له پاره اصول وضع کړي چې راجع کېږي IEEE 754 ته. دا اصول د دواړو يو ګوني (32-Bit) او دوه ګوني (64-Bit) د واضح کولو له پاره استعمالېږي. د لسو په قاعدې اعشاري اعداد په $M \times 10^E$ سره ښودل کېږي.

۱.۸.۱ یوګونې دقیقه ښــودنـه (Single Precision Representation):

اعشـاري اعداد په یو ګونې دقیقه ښـودنه کې د 32 بیټ (Bit) په واسـطه ښـودل کېږي، په ۸.۱ انځور: کې ښودل شوي.

یو اړخیز (Biased) اکسـپونینټ: دا یو اکسـپونینشـل $_2$ 127 +(01111111) دی، نو له همدی امله اکسپونینشل په مثبت عدد باندې ښودل کېږي.

نورمال مانتیزا (Mantissa) دا مانتیزا په M .1 سره ښودل کېږي، M ته نارمل مانتیزا وایي که چېرته M=00101 سره نو بیا مانتیزا 1.00101 ده.

مثـال ۱۹.۱: د $_2$(111.0000111) نـارمـل مـانتیزا او بـایزد اکســپونـنـټ پیـدا کړئ.
111.0000.111 دا هم د $1.11.0000111*2^{10}$ په شــکـل ســره لیکل کېدای شــي، چېرته چې:

M=110000111
Biased exponent = 10 + 01111111 = 10000001

په یو ګوني دقت سره 111.0000111 داسې ښودل کېږي:

M=110000111

1bit	8 bits	23 bits
0	10000001	11000011100000000000000

۸.۱ انځور: د 32 بیټ په واسطه د اعشاري اعدادو ښودل

۲۰.۱ مثال: لاندې یو ګونې دقت والا اعشاري عدد د لسو قاعدې ته تبدیل کړئ.

101111101 11001000000000000000000
S = 1 means mantissa is negative.
Biased exponent = 01111101.

Exponent = 01111101– 01111111 = -00000010.
Normalized mantissa = 11001000000000000000000.
Mantissa =1. 11001000000000000000000.
Decimal number = 1.11001000000000000000000 *210 = 0.01110011.

دوه گونی دقت: د دې له پاره چې د اعشاري اعدادو دقیق والی زیات کړو، IEEE 745 د دې له پاره دوه گونی دقت رامنځته کړو کوم چې په 64 بیټونو (Bits) سره ښودل کېږي، ۹.۱ انځور: (Biased Exponent) بایست اکسپونینټ= اکسپونینټ (1023+ (Exponent)

1bit	8 bits	23 bits
S	Biased Exponent	Normalized Mantissa

۸.۱ جدول: IEEE 745 اعشاري په یو گوني دقت سره (د مانتزا علامه ښايي =S، معنا دا چې د مانتزا علامه مثبت ده S = 0 او معنا دا چې مانتزا منفي ده S = 1)

1bit	11 bits	52 bits
S	Biased Exponent	Normalized Mantissa

۹.۱ انځور: IEEE 745 دوه گونی دقت سره د اعشاري اعدادو شکل.

جدول ۱.۴: د دوه په قاعدې د لسو قاعده بیز اعداد (BCD)

Decimal	BCD
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001

۲۱.۱ مثال: 5.75 د IEEE 745 یو ګوني دقت سره و ښایئ.

-15. 625 = $(1111.101)_2$
-1111.101 = - 1.11101101 * 2^{11}
S = 1
.Normalized Mantissa = 0.11101101
Biased Exponent = 11+ 01111111=10000010.
IEEE745 Single Precision Is
1 10000010 11101101000000000000000.

۹.۱ د دوو په قاعدې د لســو قاعده ییز اعداد (Binary-Coded (BCD))Decimal

په روزمره ژوند کې مورږ د لسو قاعدې اعدادو څخه کار اخلو لوی عدد یې ۹ دی، چې د دوو په قاعدې کې په 1001 سره ښودل کیږي. جدول ۴.۱ د لسو په قاعدې اعداد سره د دوو په قاعدې اعداد معادل ښایي.

۲۱.۱ مثال: د 345 تبدیلول د دوو قاعدې (BCD) ته

له جدول څخه کومک واخلئ: 0011 0100 0101

۲۲.۱ مثال: (10100010010) د لسو قاعدې ته تبدیل کړئ، له ښي طرف څخه چپ

طرف تـه څلور، څلور بيټونـه (Bits) جـدا کړئ، عوض کړی مربوطـه نمبر د BSD پـه واسطه چې نتيجه يې 512 په لاس راځي.

۱۰.۱ د کوډ ورکولو سيستم (Coding Schemes)

۱.۱۰.۱ اســکي کوډ (ASCII Code): هر حرف په اسـکي کوډ کې په 8 بيټونو (Bits) ســره ښــودل کېږي، چې مهم بيټ د پيريټي بيټ (Parity Bit) له پاره اسـتعمالېږي. جدول ۵.۱ اسـکي کوډ او د هغې معادل د شـپاړسـو په قاعدې سـره ښايي.

د 00 نه تر 1F پورې او 7F پورې، دې حروفو ته د شپاړسو په قاعدې سره کنټرول حروف وايي، چې نه چاپ کېدونکي حروف دي لکه NUL, SOH, STX, ETX, ESC, and DLE.

مثال ۲۳.۱: د "Network" لغت د دوو قاعدې ته تبديل کړئ او نتيجه يې د شـــپاړســـو قاعدې سره وښايئ.

د ۵.۱ جدول په استعمال سره هر حرف په 7 بيټونو (7 Bits) سره ښودل شوی دی.

1001110	1100101	1110100	1110111	1101111	1110010	1101011
N	e	t	w	o	r	k
Or in hexadecimal						
4E	**65**	**74**	**77**	**6F**	**72**	**6B**

۲.۱۰.۱ يونيکوډ (Unicode) يا نړيوال کوډ ورکولو ســيســتم (Universal Code or Unicode): نړيوال کوډ ورکولو يا يونيکوډ سيستم (Unicode) د ۱۶ بيټونو حروف - کوډ کولو نوی معياري سيستم دی، چې په ډېرو نړيوالو ژبو لکـه يونـاني، عربي، چينـايي، او جـاپـاني ژبو کې د حروفو د ښــودلو لـه پـاره استعمالېږي. د اسکي (ASCII) کوډ په لاتين کې د هر کرکټر نمايندګۍ له پاره

۸ بيټه (Bits) کاروي او دا کولی شي چې ۲۵٦ حروفو استازيتوب وکړي. اسکي (ASCII) کوډ نه شي کولی چې رياضيکې او ساينسی سمبولونه حمايه کړي، په داسې حال کې چې نړيوال يا يونيکوډ کوډ (Unicode) 16 بيټونه استعمالوي، دا کولی شي چې 65,536 حروف يا سمبولونه حمايه کړي. يو حرف په يونيکوډ کې په 16 بيټونو کې د دوو په قاعدې سره ښودل کيږي.

يا په 4 اعدادو سره د شپاړسو په قاعدې سره ښودل کيږي، د مثال په توګه د B حرف په يونيکوډ سره په U0042H داسې ښودل کيږي (U معنی يونيکوډ) او اسکي (ASCII) کوډ سره په $(00)_{16}$ او $(Ff)_{16}$ ښودل کيږي. د اسکي (ASCII) کوډ نه يونيکوډ ته د تبديلولو له پاره دوه صفرونه د اسکي کوډ چپ طرف ته اضافه کيږي، له همدې امله يونيکوډ د اسکي کوډ د ښودلو له پاره داسې دی.

Table 1.5 American Standard Code for Information Interchange (ASCII)

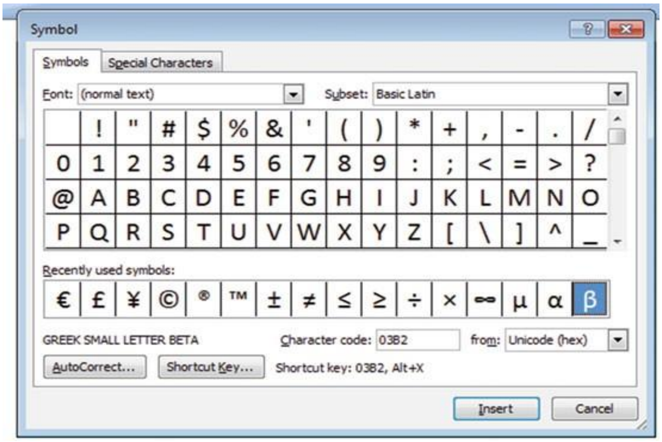
Binary	Hex	Char	Binary	Hex	Char	Binary	Hex	Char	Binary	Hex	Char
0000000	00	*NUL*	0100000	20	**SP**	1000000	40	**@**	1100000	60	,
0000001	01	*SOH*	0100001	21	**!**	1000001	41	**A**	1100001	61	**a**
0000010	02	*STX*	0100010	22	**"**	1000010	42	**B**	1100010	62	**b**
0000011	03	*ETX*	0100011	23	**#**	1000011	43	**C**	1100011	63	**c**
0000100	04	*EOT*	0100100	24	**$**	1000100	44	**D**	1100100	64	**d**
0000101	05	*ENQ*	0100101	25	**%**	1000101	45	**E**	1100101	65	**e**
0000110	06	*ACK*	0100110	26	**&**	1000110	46	**F**	1100110	66	**f**
0000111	07	*BEL*	0100111	27	**'**	1000111	47	**G**	1100111	67	**g**
0001000	08	*BS*	0101000	28	(	1001000	8	**H**	1101000	68	**h**
0001001	09	*HT*	0101001	29	)	1001001	49	**I**	1101001	69	**i**
0001010	0A	*LF*	0101010	2A	*****	1001010	4A	**J**	1101010	6A	**j**
0001011	0B	*VT*	0101011	2B	+	1001011	4B	**K**	1101011	6B	**k**
0001100	0C	*FF*	0101100	2C	,	1001100	4C	**L**	1101100	6C	**l**
0001101	0D	**CR**	0101101	2D	-	1001101	4D	**M**	1101101	6D	**m**
0001110	0E	*SO*	0101110	2E	.	1001110	4E	**N**	1101110	6E	**n**
0001111	0F	**SI**	0101111	2F	/	1001111	4F	**O**	1101111	6F	**o**
0010000	10	**DLE**	0110000	30	**0**	1010000	50	**P**	1110000	70	**p**
0010001	11	**DC1**	0110001	31	**1**	1010001	51	**Q**	1110001	71	**q**
0010010	12	**DC2**	0110010	32	**2**	1010010	52	**R**	1110010	72	**r**
0010011	13	**DC3**	0110011	33	**3**	1010011	53	**S**	1110011	73	**s**
0010100	14	**DC4**	0110100	34	**4**	1010100	54	**T**	1110100	74	**t**
0010101	15	**NACK**	0110101	35	**5**	1010101	55	**U**	1110101	75	**u**
0010110	16	**SYN**	0110110	36	**6**	1010110	56	**V**	1110110	76	**v**
0010111	17	**ETB**	0110111	37	**7**	1010111	57	**W**	1110111	77	**w**
0011000	18	**CAN**	0111000	38	**8**	1011000	58	**X**	1111000	78	**x**
0011001	19	**EM**	0111001	39	**9**	1011001	59	**Y**	1111001	79	**y**
0011010	1A	**SUB**	0111010	3A	:	1011010	5A	**Z**	1111010	7A	**z**
0011011	1B	**ESC**	0111011	3B	;	1011011	5B	[	1111011	7B	[
0011100	1C	**FS**	0111100	3C	<	1011100	5C	\	1111100	7C	\
0011101	1D	**GS**	0111101	3D	=	1011101	5D	]	1111101	7D	}
0011110	1E	**RS**	0111110	3E	<	1011110	5E	^	1111110	7E	~
0011111	1F	**US**	0111111	3F	?	1011111	5F	-	1111111	7F	**DEL**

۱.٦ جدول یونکوډ (Unicode) قیمتونه د لاتین او یونانيي حروفو له پاره

Latin		Greek	
Character	Code (hex)	Character	Code (hex)
A	U0041	φ	U03C6
B	U0042	α	U03B1
C	U0043	γ	U03B3
0	U0030	µ	U03 BC
8	U0038	β	U03B2

۷.۱ جدول یونکوډ (Unicode) د بلاک ځایونه

Start code (hex)	End code (hex)	Block name
U0000	U007F	Basic Latin
U0080	U00FF	Latin supplement
U0370	U03FF	Greek
U0530	U058F	Armenian
U0590	U05FF	Hebrew
U0600	U06FF	Arabic
U01A0	U10FF	Georgian



۱۰.۱ انځور: د یونکوډ (Unicode) مثال

جـدول ۶.۱ د $(0000)_{16}$ او $(00ff)_{16}$ پـه منځ کې بعضـــی یونیکوډ د لاتین حروفوله پاره او بعضې د یوناني له پاره را ښـایي . یونیکوډ د کوډ په بلاکونو تقسیم شوی،

هر بلاک د خپلې خاصې ژبې ته سپارل شوی. جدول ۷.۱ هر بلاک د یونیکوډ د مختلفو ژبو له پاره تشریح کوی ۱۰.۱ انځور د یونیکوډ مثال: د میکروسافټ ورډ خلاص کړئ او په انزرټ (Insert) مینو کلیک وکړئ او بیا به سمبولونه ښکاره شي لکه په ۱۰.۱تصویر په هر حرف چې کلیک وکړې نو د یونیکوډ قیمتونه به یې ښکاره شي د مثال په توګه د β یونیکوډ مساوي دی له 03B2 د شپاړسو په قاعدې کې.

۱.۱۱ پیریټي بیټ (Parity Bit)

پیرټي بیټ د غلطیو د پیژندلو له پاره کارول کیږي ، ځکه چې ممکن له سرچینې څخه منزل ته د معلوماتو د لیږد پرمهال یو بیټ یا بیټونه بدل شي ، نو د پیرټي بیټ یو اضافي معلومات دي چې د معلوماتو د لیږلو په وخت کې د معلوماتو سره یو ځای کیږي. دا ښایي چې د معلوماتو په اصلي لیږنه کې د 0 او یا 1 تعداد جفت دی او که طاق چې ددې پواسطه منزل ته یو خبر ورکول کیږي چې ایا معلومات په سمه توګه رسیدلی او کنه.

جفت پیریټي: د (صفر یا یو) پداسې حال کې چې د 1 تعداد جفت وي .

۲۴.۱ مثال: زمونږ پیغام $(00111)_2$. دی، د پیغام په چپ طرف کې د یو عدد په اضافه کولو سره له دې نه $(100111)_2$. جوړ شو. زمونږ پیریټی جفت بیټ په مجموع کې د جفت عددونو مجموعه جوړه کړه (د 3 نه تر 4 پورې).

زمونږ پیغام $(10111)_2$ دی، د پیغام په چپ طرف کې د صفر په اضافه کولو سره، مونږ سره اوس دا $(010111)_2$ جوړ شو، زمونږ جفت پیریټي بیټ د یو (1) جفت عددونه پریښودل (4 ځله د یو عددونه).

طاق پیریټي: یو اضافي بیټ (صفر یا یو) د دی له پاره انتخاب شوي چې د یو عدد له تعداد څخه طاق عدد لاسته راځي.

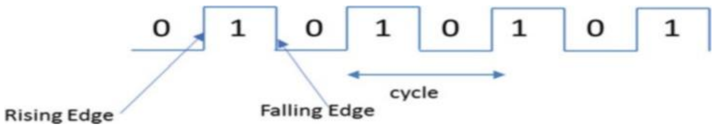
زمونږ پیغام $(10111)_2$ دی. د پیغام چپ طرف ته د یو عدد په اضافه کولو سره، دا $(110111)_2$. جوړ شو. زمونږ طاق پیریټي بیټ نه ټول د یو عددونه جفت شول (د 4 نه

تر 5 پورې د يو عددونه).

۱۲.۱ تکرار (Clock)

د 0 او 1 پـه دوامـداره توګـه تکراريـدو تـه دوران وايي لکـه څرنګـه چې پـه ۱۱.۱ انځور: ښــودل شــوې، کلـه چې دوران لـه 0 څخه 1 ته ورســيږي دې ته د دوران پورته کېدونکې څنډه (Rising Edge) وايي او کله چې دوران له 1 څخه 0 ته ورســيږي دی ته غورځيدونکي څنډه (Falling Edge) وايي .

هر دور د دوران له 1 او 0 څخه جوړ دی يا له 0 او 1 څخه چې دا په وخت (ثانيو) اندازه کيږي. کـه چيري يو دور پـه T ســره و ښــودل شــی او T ثـانيي وي نو بيـا F =1/ T (Frequency) چيري چې دفريکونســي واحد هرتز (Hz) دی اود وخت T واحد ثانيه ده.



۱۱.۱ انځور دوارنی سګنالونه

۲۵.۱ مثال: که چيري يو دور د دوران مساوي وي له 0.5 ms سره نو د دوران فريکونسي به څو وي؟

$$F = 1 = T = 1 = 0:5 \times 10—3 = 2000 \text{ Hz}$$

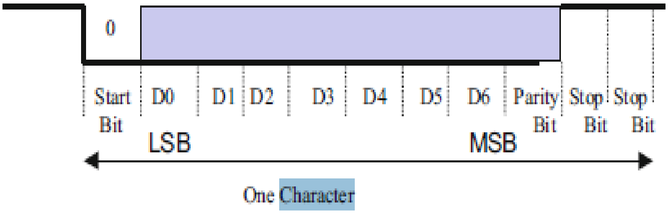
1000 Hz	Kilohertz (KHz)
10^6 Hz	Megahertz (MHz)
10^9 Hz	Gigahertz (GHz)

۱۳.۱ د لیږلو طریقې (Transmission Modes)

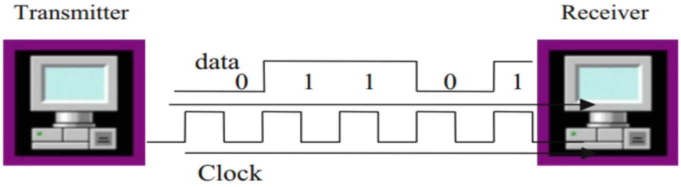
کلـه چې معلومات د ډیجیټل (Digital) ســیگنلونو له لارې له یو کمپیوټر څخه بل کمپیوټر ته لیږدول کېږي ، نو معلومات ترلاسه کونکې کمپیوټر باید د هر سیگنال اندازه تشخیص کړي تر څو معلومه کړي چې کله سیگنل پای ته رسیږي او کله بل سیگنل پیـل کېږي. د مثـال پـه توگه ، کله چې یو کمپیوټر ســیگنل لیږي لکه څنگه چې په ۱۲.۱ انځور کې ښــودل شــوي ، معلومات ترلاســه کونکې کمپیوټر باید دا معلومه کړي چې په سیگنال کې څومره 1 او 0 موجود دي. د سرچېنې او منزل مقصود وسیلو تر منځ د همغږۍ میتودونه عموما په دوه کټگوریو کې ویشـل شـوي دي: مسلسل لېږد (Synchronous) او وقفه یي لېږد (Asynchronous)

۱.۱۳.۱ وقفه یي لېږد (Asynchronous Transmission): وقفه یي لېږد حرف په حرف لیږلو تـه وایي او د پرله پســې (Serial) اړیکو له پاره اســتعمالیږي لکه موډیم یا سیریل فرنټر(Serial Printer). په وقفیي لېږدونه کې د معلوماتو هر حرف په شروع کې یو بیټ (Bit) لري چې د حرف شــروع پرې و پیژندلی شــې او یو یا دوه بیټه لري چې یو یي د حروفو د اخر د پیژنـدلو له پاره هم لیکل کیږي، لکه څنگه چې په ۱۳.۱ انځور کې ښــودل شــوی. د معلوماتو حروف اوه بیټه وی. باقي پاتي بایټونه (Bits) به پیریټي بیټ (Parity Bit) وی کوم چې استعمالیږي د

۱۲.۱ انځور ډیجیټل سیگنل



۱۳.۱ انځور وقفه يي لېږد



۱۴.۱ انځور: متواتر لېږد

د معلومات ترلاسه کونکي کمپيوټر (Receiver) په ذريعه د غلطيو پيداکېدو له پاره. وروسته کله چې پيريټي بيټ (Parity Bit) ولېږل شي، سيګنل بايد لږترلږه ۱ بيټ ته لوړ شي ترڅو د حرف اخر وپېژندل شي. نوی شروع شوی بيټ د دې له پاره استعمالېږي د ترلاسه کونکي الي له پاره د شاخص په توګه کار وکړي کوم چې د معلوماتو حروف راځي او د معلومات ترلاسه کونکي اړخ ته اجازه ورکوي چې خپل دوران جوړ کړي. لکه څنګه چې د معلومات ترلاسه کونکي او لېږونکي دوران په دوامداره ډول نه جوړېږي، نو لېږونکي د شروع بيټ استعمالوي ترڅو د ترلاسه کونکي دوران بيرته تنظيم کړي نو دا د لېږونکي له دوران سره سمون خوري. همدارنګه، د معلومات ترلاسه کونکي وار دمخه د لېږونکي لخوا لېږل شوي هر حرف کې د بيټونو تعداد شمير له پاره پروګرام شوی وي.

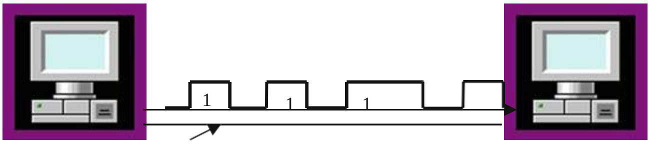
۲.۱۳.۱ مسلسل لېږد (Synchronous Transmission): بعضې پروګرامونه ضرورت لري چې يو لوی بلاک د معلوماتو ولېږی، لکه فايلونه له هارډسک نه يا د کمپيوټر نه د معلوماتو لېږل پرنټر ته. د وخت په انټروال (Interval) سره د

معلوماتو د یو لوی بلاک د لیږلو له پاره مسلسل لېږد (Synchronous) یو موثره طریقه ده.

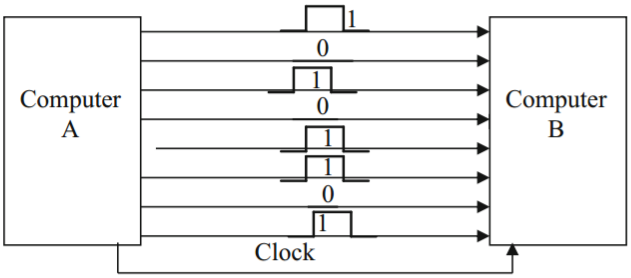
د یو بیروني ارتباط کارولو له لارې چې د دوران تغیر (Clock Pulse) رامینځته کوي د دې له پاره معلومات ترلاسه کونکې (Receiver) او مسلسل لیږونکې (Synchronous) یوه طریقه. د دوران تغیر (Clock Pulse) د سیګنل د معلوماتو کچه وړاندې کوي ،لکه څنګه چې په انځور: ۱۴.۱ کې ښودل شوي ، او د معلوماتو لېږد سرعت ټاکې. ۱۴.۱ انځور کې ترلاسه کونکي (Receiver) معلومات د ۰۱۱۰۱ په توګه پیژني ، د هر بیټ (Bit) پراخوالې د یو دوران په وسیله ښودل کیږي.
۱۴.۱ انځور راښایي چې د مسلسل (Synchronous) لېږد له پاره د دوران تغیر (Clock Pulse) انتقال له پاره یو اضافي وصلیدل ضروري دی. په نیټورک کې ، یو موډیم د دواړو معلوماتو لیږلو او د دوران تغیر (Clock Pulse) له پاره کارول کیږي. دوه سیګنالونه په داسې ډول کوډ شوی چې د متواتر سیګنال په معلوماتو کې ځای په ځای شوي. چې دا د منچیستر (Manchester) کوډ کولو یا متفاوت مانچستر کوډ کولو سره ترسره کېدای شي.

۱۴.۱ د لېږدوني طریقې (Transmission Methods)

د ارتباطي چېنل په اوږدو کې د ډیجیټل سیګنال لیږلو له پاره دوه ډوله د لېږدوني طریقې کارول کیږي: پرله پسې لېږد (Serial Transmission) او موازي لېږد (Parallel Transmission).



۱۵.۱ انځور پرله پسې لېږد



۱۶.۱ انځور موازی لېږد

۱.۱۴.۱ مسلسل لېږد (Serial Transmission): په پرله پسې لېږد کې ، معلومات پـه یو وخـت کې ۱ بیـټ د یو لین لـه لارې لېږدول کېږي لکه څنګه چې په ۱۵.۱ انځور کې ښودل شوي.

۲.۱۴.۱ موازی لېږد (Parallel Transmission): پـه موازي لېږد کې ، ډیر بیټونه په یوځل لیږل کېږي ، د پرله پســې لېږد په ځای چې بیټ په بیټ پکې لیږل کیږي پـه یو وخـت کې ۱ بایټ یا ډیر لیږل کیږي. ۱۶.۱ انځور ښــایي چې د ۸ مختلفو لینونو په ذریعه څنګه A کمپیوټر په عین وخت کې د معلوماتو ۸ بیټونه د B کمپیوټر ته لیږي، په عین دوران د ســرعت کې موازي لېږد د پرله پســې لېږد څخه ګړندی دی.

۱.۱۵ خلاصه

د دوه برقي سيسټمونو تر منځ د معلوماتو تبادله د ډيجيټل يا انلاګ سيګنلونو په بڼه ترسره کيږي.

- عصري کمپيوټرونه په ډيجيټل سيګنلونو باندی کار کوي
- ډيجيټل سيګنالونه د دوو ولټيجونو په واسطه ښودل کيږي.
- دوه ييز (Binary) اعداد د دوو په قاعدې سره ښودل کيږي.
- د سيګنال فريکونسي د وخت د دوران معکوسه ده.
- په باينری کې يو عدد ته بيټ (Bit) وايي، 8 بيټو ته بايټ (Byte) وايي او 4 بايټو (Byte) ته يو لغت وايي.
- په کمپيوټر کې دننه چې کوم معلومات دی دا ټول په باينری يا د دوو په قاعدې سره ښودل کيږي.
- منفي اعداد د کمپيوټر په دننه کې د دوه ييز کمپليمنټ (Two's Complement) په واسطه ښودل کيږي.
- په علامه لرنکې اعدادو کې مهم بيټ د يو عدد نښه يا علامه ښايي.
- مثبت عدد په صفر سره ښودل کيږي او منفي عدد په يو سره ښودل کيږي.
- يويز کمپليمنټ (One's Complement) د باينری اعدادو د هر عدد د کمپليمنټ بيټ وي.
- دوه ييز کمپليمنټ (Two's Complement) د دوه ييز عدد د جمعې يو کمپليمنټ وي.
- د دوه په قاعدې د لسو قاعده ييز اعداد (BCD) د 0 نه تر 9 پورې د لسو په قاعدې اعدادو ښودلو له پاره استعماليږي.
- د شپاړلسو په قاعدې عددی سيستم په ډيجيټل سيسټمونو او کمپيوټرونو کې د بائنري مقدارونو د نمايندګۍ مؤثريت په توګه کارول کيږي.
- پيريټي بيټ (Parity Bit) د يو بيټ د غلطيو د پيدا کيدو له پاره استعماليږي.

- د IEEE 757 په معیار د اعشاري اعدادو د ښودلو له پاره استعمالیږي.
- په کمپوټر کې دننه معلومات په اسکي کوډ (ASCII)سره ښودل کیږي، اسکې کوډ د 7 بیټونو څخه جوړ شوی.
- د کمپیوټر د اجزاو ترمینځ معلومات په پرله پسې (Serial) یا موازي (Parallel) بڼه لیږدول کیږي.
- په پرله پسې لیږد کې، په یو وخت کې 1 بیټ معلومات لیږل کیږي.
- په موازی لیږد کې، په یو وخت کې ډیر بیټونه (Bits) لیږل کیږي.

ستونزې او پوښتنې

1. د کمپیوټر د ورکړېز (Input) د درېو الو نومونه ولیکئ.
2. د کمپیوټر د راکړېز (Output) د درېو الو نومونه ولیکئ.
3. انالوګ سیګنل وښایئ
4. ډیجیټل سیګنال وښایئ
5. دا څو بیټه کیږي
   a. بایټ (Byte)
   b. نیم لغت (Half Word)
   c. لغت (Word)
6. لاندې د لسو په قاعدې (Decimal) اعداد تبدیل کړی د دوو قاعدې (Binary) ته:
   a. 35
   b. 85
   c. 23.25
7. لاندې د دوو په قاعدې (Binary) اعداد تبدیل کړی د لسو قاعدې (Decimal) اعدادو ته :
   a. 1111101
   b. 1010111.1011
   c. 11111111
   d. 10000000
8. لاندې د دوو په قاعدې (Binary) اعداد تبدیل کړی د شپاړلسو قاعدې (Hexadecimal) اعدادو ته:
   a. 1110011010
   b. 1000100111
   c. 101111.101

9. په لاندې یو دور(Cycle) د دوران سره د ډیجیټل سیګنل فریکونسي پیدا کړئ:

a. 1 S
b. 0.1 S
c. 0.02 S
d. 0.02 Ms

10. لاندې د ډیجیټل سیګنل فریکونسي در کړل شوې، د ډیجیټل سیګنل دور(Cycle) د دوران پیدا کړئ:

a. Hz
b. 200Hz
c. 10,000 Hz
d. 4 Mh

11. لاندې اعداد د لسو قاعدې ته تبدیل کړئ:

a. $(34a)_{16}$
b. $(Fac)_{16}$

12. لاندې د لسو قاعدې اعداد (Decimal) د شپاړلسو قاعدې ته تبدیل کړئ:

a. $(234)_{10}$
b. $(75)_{10}$

13. لاندې اعداد د دوو قاعدې (Binary) ته تبدیل کړئ:

a. $(3FDA)_{16}$
b. $(FDA.5F)_{16}$

14. لاندې جمع اجرا ترسره کړئ:

$$\begin{array}{r} 1101010 \\ +\ 1010111 \\ \hline \end{array} \qquad \begin{array}{r} 1101010 \\ 1011011 \\ \hline \end{array}$$

15. د لاندې اعدادو دوه ییز کمپلیمنټ (two's complements) پیدا کړئ:

a. 11111111
b. 10110000
c. 10000000

d. 00000000

۱۶. د "LOGIC" لغت لاندې معنی ورکوي

a. په اسکي (ASCII) کې ښودل

b. هر حرف ته جفت پیریټي بیټ اضـافه کړئ او د شپاړلس قاعدې په واسطه یې وښایئ:

۱۷. $(465)_{10}$ دا بی سی ډی(BCD) کې وښایئ:

۱۸. $(100101100111)_{BCD}$ د لسو په قاعدې (Decimal) کې وښایئ:

۱۹. $(110010000100)_{BCD}$ د لسو په قاعدې (Decimal) کې وښایئ:

20. لاندې دوه ییز کمپلیمنټ (Two's Complement) اعداد د لســو په قاعدې (Decimal) کې وښایئ:

a. 1011

b. 11111001

c. 10011111

۲۱. لانـدې بی علامی لرونکي اعـداد یو لـه بل څخه د دوه ییز کمپلیمنټ (Two's Complement) په ذریعه منفي کړی:

a. 11110011–11000011

b. 10001101–11111000

۲۲. لاندې د علامه لرونکي اعدادو جمع اجرا کړی، فرض کړی چې هر عدد 6 بیټه ده او دا هم بیان کړی چې که د جمع نتیجه زیاته (Overflow) نه شي:

a. $(+12) + (+7)$

b. $(+25) + (+30)$

c. $(-5) + (+9)$

d. $(-6) + (-7)$

e. $(-36) + (-12)$

۲۳. کوم یو غـټ 16 بیټـه بـاینری (Binary) قیمت ده کوم چې په یو د لاندې ســره ښودل کیږي:

a. Unsigned number

b. Signed magnitude

c. Signed two's complement

۲۴. لاندې د لسـو قاعدې (Decimal) اعداد تبدیل کړی د 745 Ieee اصـولو په یوګونی دقت (Single Precision) سره:

a. 34.375

b. -0.045

۲۵. لاندې د IEEE 745 اصـولو د یوګونی دقت (Single Precision) د لسـو قاعدې (decimal) اعدادو ته تبدیل کړی:

a. 1 10000100 01110000000000000000000

b. 01111100 11100000000000000000000

26. د لېږد د طریقو ډولونه ولیکئ:

27. لانـدې علامـه لرونکې میګنیچوډ (Magnitude) اعـداد تبـدیـل کړی د لسـو قاعدې (Decimal) ته:

a. 11000011

b. 10001111

۲۸. هر یو د لانـدې اعـدادو د 8 بیټو پـه علامه لرونکي دوه ییز کمپلیمنټ (Two's Complement) کې وښایئ:

a. -15

b. -24

c. -8

۲۹. لاندې جمع اجرا کړی:

a. 0F4A

b. +420B

دویم څپرکی

بولیین منطق او منطقي ګیټونه
(Boolean Logics and Logic Gates)

موخې: د دغه څپرکي په پای کې، به تاسو وکولئ شی چې:

- د بولین تیورمس (Boolean Theorems) په ساده عملیو وپوهیږئ.
- د مختلفو منطقي ګیټونو(Logic Gates) لکه انډ(AND) ، ار(OR) ،ناټ (NOT) ایکس ار(XOR) او ننډ (NAND) عملیې څرګندې کړئ.
- د مختلفو منطقي ګیټونو (Logic Gates) صدق جدول (Truth Table) ښکاره کړای شئ.
- د مختلفو مدغم شوو سرکټونو((Integrated Circuits (ICs)) تر منځ توپیر وکولای شئ.
- د بولین تابع ګانو (Boolean Functions) د اسانه کولو له پاره بولین تیورمس (Boolean Theorems) تطبیق کړئ.
- د بولین تابع ګانو (Boolean Function) له پاره منطقي سرکټ (Logic Circuit) رسم کړئ.

- د بولین تـابع (Boolean function) لـه پـاره صــدق جدول (Truth Table) ښکاره کړئ.
- د ډیجیټل منطقي سرکټ (Digital Logic Circuit) له پاره راکړیزه (Output) تابع (Function) پیدا کړئ.
- د LSI, MSI, SSI او VLSI تر منځ توپیر وکولای شئ.

۱.۲ پیژندنه

منطقي ګیټونه (logic Gates) دټرانزیســټورونو (Transistors) مجموعه ده، چې د مدغم شــوي ســرکټ (Integrated Circuit) بنیادي اجزآ تشــکیلوي. منطقي ګیټونه (Logic Gates) د ډیجیټل سیستم د ډیزاین کولو له پاره استعمالیږي، چې درې بنیـادي عملیې لري عبـارت دي لـه :انـډ (AND)، ار (OR) او نـاټ (Not) څخه. د ډیجیټل سیستم ځانګړنې د تابع (Function) اویا هم د صدق جدول (Truth Table) پـه واســطـه ښــودل کـیږي. بـولـیـن تیورمس (Boolean Theorems) د بولین تابع (Boolean Function)داسانه کولو له پاره استعمالیږي، په دې ترتیب چې پکې لږ تعـداد منطقي ګیټونـه (Logic Gates) اســتعمـال شــي. مـدغم شــوی ســرکـټ (Integrated Circuit) پـه هغه کې د موجوده ګیټونو(Gates) د تعداد له مخې طبقه بندي کیږي، چې عبارت دي له: SSI, MSI, LSI او VLSI څخه.

۲.۲ بولین منطق او منطقي ګیټونه (Boolean logic and logic gates)

انډ منطق (And Logic): انډ منطق (AND Logic) د یو ټکي "." په واسطه ښــودل کیږي، اکثره وخـت کولای شــو چې د (And Logic) بغیر لـه ټکي څخه هم ولیکو. (X.Y) یا (XY) چې د (X) انډ(AND) (Y) په ډول تلفظ کیږي.

پـه $X\ AND\ Y = Z$ یـا $(X.Y = Z)$ کې د Z قیمت هغه وخت یو $(Z = 1)$ وي

کله چې د X او Y قیمتونه له یو سره مساوي وي یعنې $X\ AND\ Y = 1$ یا $X.Y = Z$ => $X = 1\ ,Y = 1$ که د X یا Y یا د دواړو قیمتونه صفر شي نو $Z = 0$ دی.

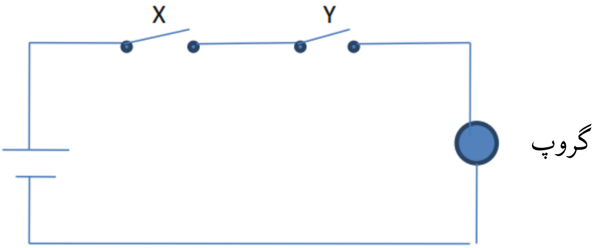
د انډ منطق (AND Logic) د ۱.۲ انځور په څیر د بریښنایي (Electrical) سرکټ په واسطه هم ښودل کیږي. فرض کړئ چې X او Y دوه سویچونه (Switches) او Z یو ګروپ دی، $X = 0$ او $Y = 0$ یعنې سویچونه دواړه خلاص یا (Open) دي او ګروپ بند یا Off دی که چیرې ګروپ Off وې $Z = 0$ او که چیرې ګروپ ON وي $Z = 1$ دی نو موږ یو جدول ترتیبوو ۱.۲ جدول د ۱.۲ انځور د انډ (AND) ګیټ (Gate) عملیه څرګندوي.

۲.۲ انځور د انډ (AND) ګیټ (Gate) له پاره دوه ورکړیز (2-Input) ښکاره کوي او د (2.2) جدول د انډ (AND) ګیټ (Gate) له پاره صدق جدول (Truth Table) څرګندوي ، چې هغه وخت د انډ (AND) ګیټ (Gate) راکړیز (Output) یو وي کله چې د دواړو ورکړیزو (Inputs) قیمت یو وي.

اړ منطق (OR logic): د اړ (OR) عملیه (Operation) د جمع (Plus) "+" او د V علامو په واسطه ښودل کیږي، چې د جمع (Plus) علامه د اړ (OR) عملیې له پاره تر ټولو مشهوره نښه (Symbol) ده. $X + Y$ د X اړ (OR) Y په شکل تلفظ کیږي.

په $X + Y = Z$ کې $Z = 1$ که چیرې $X = 1$ یا $Y = 1$ یا دواړه $X = 1$ او $Y = 1$ وي.

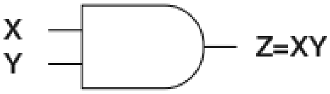
د اړ (OR) عملیه د ۳.۲ انځور په څیر د بریښنایي سرکټ (Electrical Circuit) په واسطه هم ښودلی شو. په ۳.۲ انځور کې ګروپ هغه وخت بند یا Off وي چې دواړه سویچونه بند یا Off وي او ګروپ هغه وخت روښانه یا On وي کله چې حد اقل یو سویچ هم روښانه یا On وي ۴.۲ انځور د اړ (OR) ګیټ (Gate) له پاره 2-Input او د (۳.۲) جدول د 2-Input لرونکې اوړ (OR) ګیټ (Gate) له پاره صدق جدول (Truth Table) څرګندوي.



۱.۲ انځور: د انډ (AND) عملي ښرګندول

X	Y	Light
Off	Off	Off
Off	On	Off
On	Off	Off
On	On	On

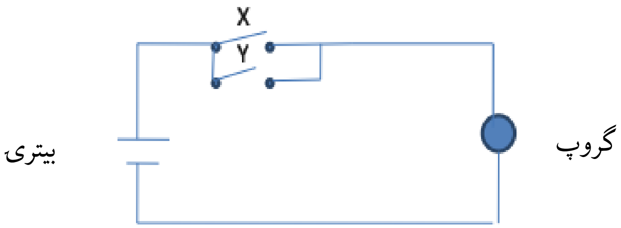
۱.۲ جدول: د۱.۲ انځور: عملیه



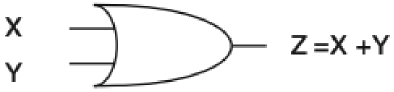
۲.۲ انځور: 2-Input انډ (AND) ګیټ(Gate)

X	Y	Z
0	0	0
0	1	0
1	0	0
1	1	1

۲.۲ جدول: د انډ (AND) ګیټ (Gate) صدق جدول(Truth Table)



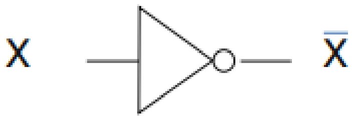
۳.۲ انځور: د ار ((OR) ګیټ (Gate) بریښناني سرکټ ځرګندونکی



۴.۲ انځور: د 2-Input ار (OR) ګیټ (Gate)

X	Y	Z
0	0	0
0	1	1
1	0	1
1	1	1

۳.۲ جدول د 2-Input ار (OR) ګیټ (Gate) د صدق جدول (Truth Table)



۵.۲ انځور: ناټ (NOT) ګیټ (Gate)

X	X′
0	1
1	0

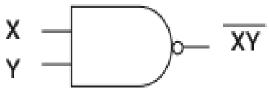
۴.۲ جدول دناټ (NOT) ګيټ (Gate) صدق جدول (Truth Table)

ناټ (NOT) منطق (logic): د ناټ NOTګيټ (gate) (Complement) تر سره کوي يعنې وايو (1) په صــفر (0) او صــفر (0) په يو (1) تبديلوي، چې همدارنګه د سرچپه کوونکي (Inverter) په نوم هم ياديږي .د Xناټ (NOT) د X` يا $\bar{X}$ په انځور ښودل کيږي.

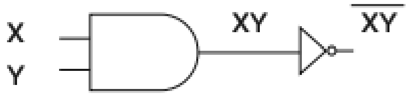
۵.۲ انځور ناټ (NOT) ګيټ (Gate) او ۵.۲ جدول د ناټ (NOT) ګيټ (Gate) صدق جدول (Truth Table) ښرګندوي.

ننډ (NAND) ګيټ (Gate): ۶.۲ انځور د لرونکې ننډ (NAND) ګيټ (Gate) ښــکاره کوي. ننډ (NAND) ګيټ (Gate) د انډ (AND) او ناټ (NOT) ګيټونو د ترکيب څخه ترلاسه کيږي. چې په ۷.۲ انځور کې ښودل شوې ، د ۵.۲ جدول د 2-Input ننډ (NAND) ګيټ (Gate) صدق جدول (Truth Table) ښرګندوي.

نار(NOR) ګيټ (Gate): ۸.۲ انځور د نار(NOR) منطق (Logic) ګيټ (Gate) ښکاره کوي، نار (NOR) ګيټ (Gate) د ار (OR) او ناټ (NOT) ګيټونو (Gates) څخه جوړشوی دی ۶.۲ جدول د 2-Input نور (NOR) ګيټ (Gate) صدق جدول (Truth Table) ښرګندوي.



۶.۲ انځور: 2-Input لرونکې ننډ (NAND) ګيټ (Gate)



۷.۲ انځور د انډ (AND) او ناټ (NOT) ګېټونو (Gates) ترکیب دی چې د ننډ (NAND) په څیر کار کوي.

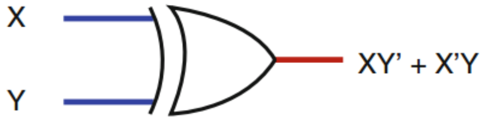
X	Y	$\overline{XY}$
0	0	1
0	1	1
1	0	1
1	1	0

۵.۲ جدول: 2-Input لرونکې ننډ (NAND) ګیټ دصدق جدول

۸.۲ انځور: نار (NOR) ګیټ

X	Y	X + Y
0	0	1
0	1	0
1	0	0
1	1	0

۶.۲ جدول: د صدق جدول د 2 راکړیز نار (NOR) ګیټ له پاره



۹.۲ انځور: 2-Input ایکس ار (XOR) ګیټ (Gate).

X	Y	X ⊕ Y
0	0	0
0	1	1
1	0	1
1	1	0

۷.۲ جدول: د ايکس ار (XOR) ګيټ (gate) صدق جدول (Truth Table).

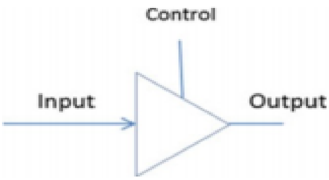
ايکسکلوسيف ار(Exclusive OR) ګيټ: ۹.۲ انځور يو ايکسکلوسيف ار(Exclusive OR) ښکاره کوي. ايکسکلوسيف ار(Exclusive OR) د ⊕ علامې په واسطه ښودل کيږي او د (XOR)په انځور ليکل کيږي. ۷.۲ جدول د ايکس ار (XOR) له پاره صدق جدول (Truth Table) ښکاره کوي.

$$X \oplus Y = X'Y + XY'$$

۱۰.۲ انځور: ايکسکلوسيف نار (Exclusive NOR) ګيټ (Gate)

X	Y	X ⊙ Y
0	0	1
0	1	0
1	0	0
1	1	1

۸.۲ جدول: د ايکسکلوسيف نار (Exclusive NOR) ګيټ د صدق جدول



۱۱.۲ انځور 3- حالته 3- State آله (Device)

ایکسکلوسیف نار (Exclusive NOR) ګیټ (gate): ۱۰.۲ انځور یو ایکسکلوسیف نار (Exclusive NOR) ګیټ (Gate) ښکاره کوي. ایکسکلوسیف نار (Exclusive NOR) د ⊙ علامې په واسطه ښودل کیږي چې د $XNOR$ په انځور لیکل کیږي. او ۸.۲ جدول او ایکسکلوسیف نار (Exclusive NOR)) ګیټ (Gate) له پاره صدق جدول (Truth Table) ښکاره کوي.

دري کاره اله (Tri State Devices): ۱۱.۲ انځور د -حالت 3-State لرونکې الې (Device) لـه پاره دیاګرام ښکاره کوي. د کنټرول لاین د دغې الې ټولې عملیې کنټرولوي. په ۱۱.۲ انځور کې که چیرې کنټرول لاین (0) ته کېښودل شي نو د ورکړېز (Input) او راکړېز (Output) تر منځ به هیڅ اړیکه موجوده نه وي. که چیرې کنټرول لاین (1) ته کېښودل شي نو د راکړېز (Output) قیمت به دورکړېز (Input) د قیمت سره مساوي وي.

متعدد ورکړېز منطقي ګیټ (Multiple input logic gate): ۱۲.۱ انځور د 3-Input لرونکې انډ (AND) ګیټ (Gate) او د هغه صدق جدول (Truth Table) ښکاره کوي.

۱۳.۲ انځور د3-Input لرونکې ار (OR) ګیټ (Gate) اود هغه صدق جدول (Truth Table) ښکاره کوي.

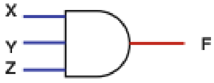
۱۴.۲ انځور3-Input لرونکې نار (NOR) ګیټ (Gate) اود هغه صدق جدول (Truth Table) روښانه کوي.

۱۵.۲ انځور 3-Input لرونکي ننډ ($NAND$) ګیټ (Gate) او د هغه مربوطه (Corresponding) جدول (Table) روښانه کوي.

۳.۲ مدغم شوي سرکټ (Integrated circuit) طبقه بندي

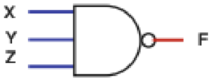
ټرانزیسټر (Transistor) د مدغم شوي سرکټ (Integrated Circuit) اساسي اجزا تشکیلوي. ۱٦.۲ انځور یو ټرانزیسټر (Transistor) او مدغم شوی سرکټ (Integrated Circuit) څرګند کوي. ټرانزیسټر (Transistor) په مدغم شوي سرکټ (Integrated Circuit) کې د سویچ په څیر عمل کوي. چې یو ای سی (IC) د سلو څخه تر میلیونونو ټرانزیسټرو (Transistor) څخه جوړیږي.

مدغم شوي سرکټونه (Integrated Circuits) په هغه کې د موجوده ګیټونو د تعداد په اساس طبقه بندي کیږي چې عبارت دي له: LSI,MSI,LSI او VLSI څخه.



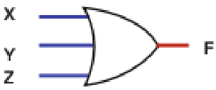
X Y Z	F
0 0 0	0
0 0 1	0
0 1 0	0
0 1 1	0
1 0 0	0
1 0 1	0
1 1 0	0
1 1 1	1

۱۲.۲ انځور 3-Input لرونکي انډ (AND) ګیټ (Gate) او د هغه صدق جدول



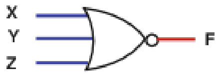
X Y Z	F
0 0 0	1
0 0 1	1
0 1 0	1
0 1 1	1
1 0 0	1
1 0 1	1
1 1 0	1
1 1 1	0

۱۳.۲ انځور: 3-Input لرونکي ار (OR) ګیټ (Gate) او د هغه صدق جدول



X	Y	Z	F
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

۱۴.۲ انځور: 3-Input لرونکې ننډ ($NAND$) ګیټ (Gate) او د هغه صدق جدول



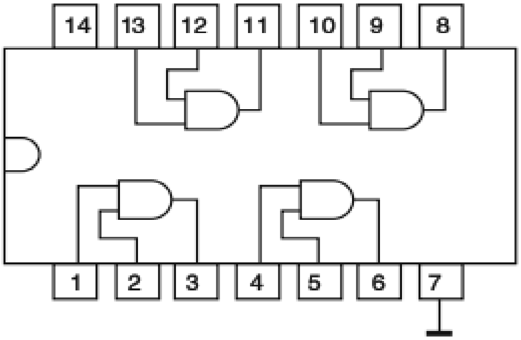
X	Y	Z	F
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

۱۵.۲ انځور: 3-Input نار (NOR) ګیټ (Gate) او د هغه صدق جدول

کوچنۍ درجه اینټیګریشن Small-Scale Integration (SSI)

SSI هغه مدغم شوي سرکټ (Integrated Circuit) ته ویل کیږي چې د لسو (10) څخه لږ ګیټونه ولري. ۱۷.۲ انځور د $74HC08$(7408) مدغم شوي سرکټ (Integrated Circuit) داخلي جوړښت ښکاره کوي، او ۱۸.۲ انځور د $74HC08$ مدغم شوی سرکټ (Integrated Circuit) انځور روښانه کوي.

۱٦.۲ انځور: (کیڼ) ټرانزیستر (Transistor) ښایي



۱۷.۲ انځور: $TTL\ 7408$ او 2-Input لرونکې انډ (AND)ګیټ (Gate)

۱۸.۲ انځور: $74LS08$ستنې(Pins)

د مدغم شوي سرکټ د ستنو شمیرنه (Integrated Circuit Pins Numbering) په ۱۸.۲ انځور کې یو $TTL\ 7408\ IC$ ښودل شوی دی یو ای سي (IC) چپ به داسې شکل سره کېښودل شي چې سرکې ګردنه (Notch) یي کېڼ لاس طرف ته شي.

د ستنو نښه کول (**Labeling**): یا نمبر ورکول د لاندېنی چپ طرف ستنې څخه چې د نچ لاندې موقعیت لري شروع او د ساعت د ستنو د حرکت په څیر پر مخ وړل کیږي.
لاندینی چپ طرف Pin د Pin لمبر 1 $(Pin\ \#1)$ په نوم یادیږي.
لاندینی ښي طرف Pin د Pin لمبر 7 $(Pin\ \#7)$ په نوم یادیږي.
پاسنی ښي طرف Pin د Pin لمبر 8 $(Pin\ \#8)$ په نوم یادیږي.
پورتنی چپ طرف Pin د Pin لمبر 14 $(Pin\ \#14)$ په نوم یادیږي.

که چېرې په ۱۸.۲ انځور کې د IC شـــمیره $74LS08$ چېرته چې LS هغه مواد ځرګندوي چې IC ور څخه جوړه شـــوي. همدارنګه په IC کې د M توري د Motorola څخه چې د IC تولیدونکې دی اخیستل شوی، Intel کمپني د " i " توري او د Texas وسایل د Texas نقشي لرونکي وي.

منځنۍ درجه اینټیګریشن (MSI) Medium-Scale Integration :

هغه IC ته ویل کېږي چې د لســو (10) او ســلو (100) تر منځ ګیټونه ولري. لکه Decoder او Multiplexer .

لوی درجه اینټیګریشن ((LSI) Large Scale Integration):

هغه IC ته ویل کېږي چې د سلو (100) او زرو (1000) تر منځ ګیټونه ولري.

ډیر لوی درجه اینټیګریشـــن (Very-large-Scale Integration): هغه IC ته ویل کېږي چې د زرو (1000) څخه زیات ګیټونه ولري.

۴.۲ بولین الجبرا تیورمس (Boolean Algebra Theorems)

بولین (Boolean) تیورمس (Theorems) د بولین تابع (Boolean Function) د اسانولو له پاره استعمالېږي په دې ترتیب چې پکې لږ ګیټونه استعمال شي. هر متحول (Variable) لکه X په باینري (Binary) کې کولی شي چې یواځې (1) یو او (0) صفر قیمت اخلي.

لومړی قانون

$$X + X = X$$

ثبوت: که چېرې $x = 0$ نو $0 + 0 = 0$ ، که چېرې $x = 1$ وي، نو $1 + 1 = 1$

نتیجه:.

$$x + x = x$$

دویم قانون

$$X + 1 = 1$$

ثبوت: که چېرې $x = 0$ نو $1 = 1 + 0$ ، $1 = x$ وي، نو $1 = 1 + 1$ ، دواړو حالتونو کې نتیجه $x + 1 = 1$ کیږي.

دریم قانون

$$X + 0 = X$$

ثبوت: که چېرې $x = 0$ وي، نو $0 = 0 + 0$، که چېرې $x = 1$ وي نو $1 = 0 + 1$ کیږي. نتیجه هغه څه کیږي چې x وي.

څلورم قانون

$$X + X' = 1$$

ثبوت: کـه چېرې $x = 0$ وي نو $1 = 1 + 0$، که چېرې $x = 1$ وي نو $1 = 0 + 1$ کیږي یعنې په دواړو حالاتو کې نتیجه مساوي کیږي له 1سره.

پنځم قانون

$$X * X = X$$

ثبوت: که چېرې $x = 1$ وي نو $1 = 1 * 1$، که چېرې $x = 0$ وي نو $0 = 0 * 0$ په دې صورت $X * X = X$ دی.

X	Y	Z	Y+Z	X(Y+Z)	XY	XZ	XY + XZ
0	0	0	0	0	0	0	0
0	0	1	1	0	0	0	0
0	1	0	1	0	0	0	0
0	1	1	1	0	0	0	0
1	0	0	0	0	0	0	0
1	0	1	1	1	0	1	1
1	1	0	1	1	1	0	1
1	1	1	1	1	1	1	1

۹.۲ جدول د $x(y+z) = xy + xz$ له پاره صدق جدول

شپږم قانون

$$X * 1 = X$$

ثبوت: کـه چیرې $x = 1$ وی نو $1 * 1 = 1$ کیږي ، کـه چیرې $x = 0$ شـي نو $0 * 1 = 0$ سره په دی صورت $x * 1 = x$ کیږي.

اوم قانون

$$X * X' = 0$$

ثبـوت: کـه چېرې $X = 0$وي نو $0 * 1 = 0$ کیږي، کـه چېرې $X = 1$شـي نو $0 = 0 * 1$ کېږي یعنې په دواړو حالاتو کې X مساوي کېږي له صفر (0) سره.

اتم قانون

$$(X')' = X$$

$(0')' = (1)' = 0\ ,(1')' = (0)' = 1$ پـه دې صــورت کې نتېجـه تل د X د قیمت سره مساوي کیږي.

توزیعي قانون(Distributive Theorem)

$$X(Y + Z) = XY + XZ$$

د پورتني خاصیت د ثبوت له پاره د تیورم د دواړو خواوو له پاره صدق جدول (Truth

(Table) په ۹.۲ جدول کې ښودل شوی، چې روښانه کوي دواړه خواوې یو شان صدق جدول (Truth Table) جوړوي.

X	Y	X+Y	(X +Y)´	X´	Y´	X´Y´
0	0	0	1	1	1	1
0	1	1	0	1	0	0
1	0	1	0	0	1	0
1	1	1	0	0	0	0

۱۰.۲ جدول د De Morgan's د قانون صدق جدول

د De Morgan لومړی قانون

$$(X + Y)' = X'Y'$$

پـه دغه قانون کې د X او Yتر منځ د اور (OR) عمليه نفي کيږي او د (OR) عمليه په انډ (AND) باندې تبديليږي.

ثبوت: د دغه تيورم د مســاوات د دواړو خواوو له پاره صــدق جدول (Truth Table) جوړوو، په نتيجه کې څرګنديږي چې د مســاوات دواړه خواوي يو شــان صــدق جدول (Truth Table) تشکيلوي.

د De Morgan دويم قانون

$$(XY)' = X'Y'$$

پـه دغه تيورم کې لومړی دواړه X او Y (Complement) کيږي او بيا د انډ (AND) عمليه په اور (OR) عمليې تبديليږي کمپليمنټ د هري برخي سره.

مثال: (WXYZ) = W’ + X’ + Y’ + Z’

د دواړو خواوو د صــدق جدول (Truth Table) په جوړولو ســره دا څرگندیږي چې نتیجه کې دواړه خواوې یو شان صدق جدول تشکیلوي.

تبدیلي قانون

$$X + Y = Y + X$$
$$XY = YX$$

اتحادي خاصت

$$X(YZ) = (XY)Z$$
$$X + (Y + Z) = (X + Y) + Z$$

د مورس (More’s) قانون

لاندني گټور قوانین Theorems دي:

a. $X + X'Y = X + Y$

b. $X' + XY = X' + Y$

c. $X + X'Y' = X + Y'$

d. $X' + XY' = X' + Y'$

مثال: لاندنی تابع گانی (Functions) ساده کړئ

(a) F(X, Y, Z) = XY’Z + XY’Z’ + XY
F(X, Y, Z) = XY’(Z+ Z’) + YZ Where Z + Z’ + 1 Then
F(X, Y, Z) + XY’þ XY + X(Y + Y’) + X

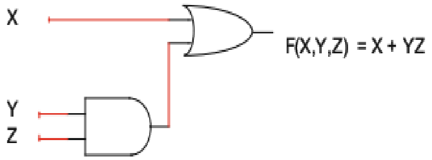
(b) F(X, Y, Z) = (X’ + Y) (X Þ Y’) + X’X + X’Y’ + XY + YY’ Where X’X And YY’ Are Zero Then F(X, Y, Z) = X’Y’ + XY

۵.۲ بولیین (Boolean) تابع (function)

بولین (Boolean) تابع (Function) د الجبري معادلاتو په واسطه چې د دوه ییز (Binary) متحولونو (Variables) لکه Y, X او Z او د دوی تر منځ منطقي (Logic) عملیې لکه انډ (AND)، اور (OR) او ناټ (NOT) څخه جوړ شوي وي، ښودل کیږي.

$F\ (X, Y, Z)\ =\ X + YZ$ یو بولین (Boolean) تابع (Function) دی.

۱۹.۲ انځور د F تابع (Function) څرګندوي چې Z او Y، X درې ورکړیز Inputs او F راکړیز (Output) ده.۱۱.۲ جدول د F تابع (Function) له پاره صدق جدول (Truth Table)روښانه کوي. صدق جدول(Truth Table) د Fتابع (Function) خصوصیات څرګندوي ، چې په دغه تابع (Function) کې $F\ =\ 1$ سره که چیرې د تابع له پاره ورکړیز Inputs $(100), (101), (110)$ او (111) وي.



۲۰.۲ انځور: د $F(X, Y, Z)\ =\ X + YZ$ تابع له پاره منطقي سرکټ.

X	Y	Z	YZ	X + YZ
0	0	0	0	0
0	0	1	0	0
0	1	0	0	0
0	1	1	1	1
1	0	0	0	1
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

۱۱.۲ جدول د $F\ (X, Y, Z)\ \ = X + YZ$ تابع (Function) له پاره صدق جدول

د تـابع (Function) کمپلیمنـټ (Complement): د یو تـابع (Function) د کمپلیمنـټ (Complement) لـه پـاره بـایـد د تـابع (Function) دواړه خواوې کمپلیمنټ (Complement) شي.

مثال: لاندې تابع (Function) کمپلیمنټ (Complement) کړئ.

$$F(X,Y,Z) = XY + Y'Z$$

د تابع د دواړو خواوو په کمپلیمنټ (Complement) کولو سـره نتیجه داسـې لاسـته راځي چې:
د ډی مورګانز (De Morgan's) د تیورم استعمال کړی

$$F'(X,Y,Z) = (XY + Y'Z')'$$
$$F'(X,Y,Z) = (XY)'(Y'Z)'$$
$$F'(X,Y,Z) = (X' + Y')(Y + Z')$$

مثال: د لاندې تابع (Function) کمپلیمنټ (complement) لاسته راوړئ.

$$F(X,Y,Z) = (X' + Y')(Y + Z')$$

د تابع (Function) دواړه خواوې کمپلیمنټ (complement) کوو:

$$F'(X,Y,Z) = [(X' + Y')(Y + Z')]'$$

په نتیجه کې د De Morgan تیورم په تطبیق کولو سره

$$F'(X,Y,Z) = [(X' + Y')] + [(Y + Z')]'$$
$$F'(X,Y,Z) = [(XY)] + [(Y'Z)]$$

$$F'(X,Y,Z) = (XY) + (Y'Z)$$

٦.٢ خلاصه

- بولين (Boolean) منطق (Logic) د انډ (AND)، ار (OR) او ناټ (NOT) منطق(Logics) څخه تشکيل شوی دی.
- د 2-Input لرونکې انډ (AND) ګيټ (Gate) راکړيز (Output) مساوي کيږي د يو(1) سره کله چې دواړه ورکړيز (Inputs) يو(1) وي ،بغير له دی صورت څخه راکړيز(Output) مساوي کېږي د صفر(0) سره.
- د 2-Input لرونکې ار (OR) ګيټ (Gate) راکړيز (Output) مساوي کيږي د يو(1) سره حداقل کـه چېرې يواځې يو ورکړيز Input هم يو(1) وي، که نه ورکړيز(Output) مساوي کيږي د صفر سره.
- ناټ (NOT) ګيټ(Gate) د one's complement عمليه تر سره کوي.
- مدغم شوي سرکټ (Integrated Circuit) په SSI (Small-Scale Integration)
- LSI (Large-Scale، MSI (Medium-Scale Integration) Integration) او VLSI (Very-Large-Scale Integration) باندی ويشل شوي دي.
- د ننډ (NAND)ګيټ (Gate) معادل ده انډ-ناټ (AND-NOT) ګيټ (Gate) سره.
- د نور (NOR) ګيټ (Gate) معادل ده د ار-ناټ (OR-NOT) ګيټ(Gate) سره.
- د 2-Input ايکس-ار(XOR) ګيټ (Gate) راکړيز (Output) مساوي کيږي د يو(1)سره کله چې دواړه ورکړيز Inputs سره مساوي نوي، که نه (Output) مساوي کيږي د صفر (0) سره.
- 2-Input لرونکې ايکسکلوسيف نار (XNOR) (Exclusive NOR) معادل ده XOR-NOT سره.

پوښتنې

۱. د $4-input$ لرونکې انډ (AND)،ار (OR)، نار(NOR) او ننډ (NAND) ګیټ له پاره Truth Table جوړکړئ.

۲. که چېرې $A = 11001011$ او $B = 10101110$ سره نو د لاندې عملیو نتیجه لاسته راوړئ

a. A AND B

b. A OR B

۳. که چېرې $A = 11001011$ سره او $B = 10101110$ سره، د لاندې عملیو نتیجه لاسته راوړئ.

(F په هیګزا (Hexa Decimal) مساوي دی د 1111 سره).

a. A NOT

b. A XOR B

c. A AND 0F

d. A AND F0

۴. د لاندې تابع (Function) له پاره منطقي (Logic) سرکټ رسم کړئ.

a. F(X,Y,Z) = Xy’ + Xy’ +Xy

b. F(X,Y,Z) =(X+Y’)(X’+Y’) (X+Y)

۵. لاندې معادلات دبولین (Boolean) تابع ګانو (Functions) څخه په استفادې ساده کړئ.

a. X + X + X

b. Xy + Xy

c. Yyy

d. X + Xy

e. Xy’ + Y’

f. (X + Y)Y’

g. (Xy) + (Xy)’

h. X’y’ + Xy

۶. لاندې تابع ګانی (Functions) ساده کړئ.

a. F(X,Y,Z) = Xy + X’y + Xz
b. (X,Y,Z) = (X + Y)(X’ + Y +Z)
c. F(X,Y,Z) = XY’Z + XYZ + Y’Z
d. F(X,Y,Z) = XY + X’YZ
e. F(X,Y,Z) = X’Y + XYZ’
f. F(X,Y,Z) = (XY) + (X+Y+Z)’X + YZ
g. F(X,Y,Z) = (XY)’ + (X+Y+Z)’

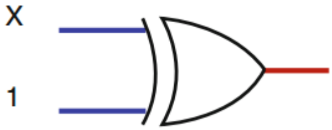
۷. د لاندې تابع ګانو (Functions)له پاره صدق جدول پیدا کړئ.

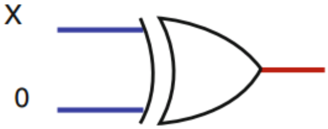
a. F(X,Y,Z) = Xy’ + Yz + Xz’
b. F(X,Y,Z) = (X+Y’)(Y+Z)(X’+Z’)

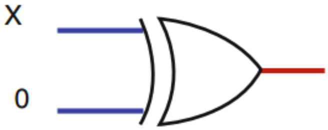
۸. که چېرې $A = 10110110$ او $B = 10110011$ سره نود لاندې عملیو نتیجه پیدا کړئ.

a. A NAND B
b. A NOR B
c. AXOR B

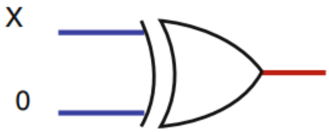
۹. د لاندې ګیټونو حاصل (Output) لاسته راوړئ.

a)


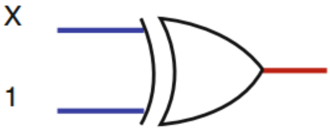
b)


c)


d)

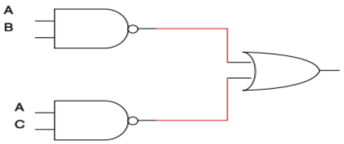


e)

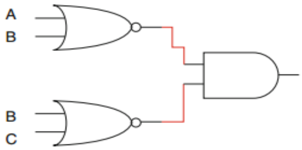


۱۰. د لاندې منطقي (Logic) سرکټ حاصل (Output) لاسته راوړئ.

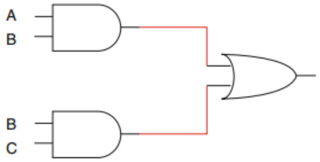
a)



b)

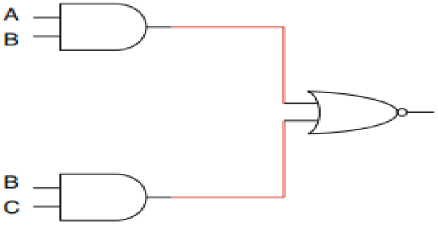


c)

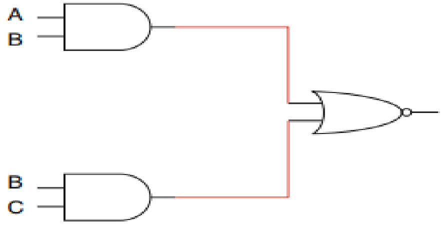


۱۱. د لاندې منطق (Logic) سرکټ حاصل تابع ګاني (Functions) لاسته راوړئ.

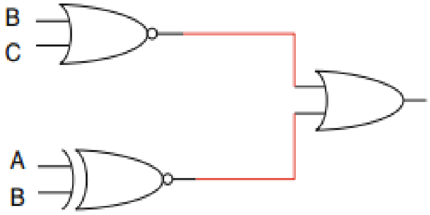
a)



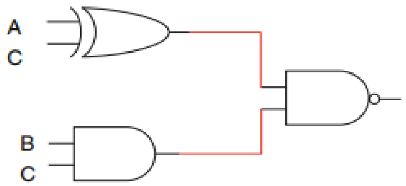
b)



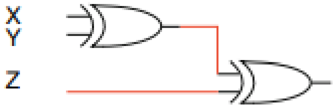
c)



d)



۱۲. د لاندې منطق (Logic) سرکټ حاصل تابع (Function) لاسته راوړئ.



۱۳. د لاندې تابع ګانو (Functions) له پاره منطق (Logic) سرکټ او صدق جدول لاسته راوړئ.

a. $F(X,Y) = (Xy)' + X(Z + Y')$

b. $F(X,Y,Z) = (X+Y+Z')'(X' + Y')$

c. $F(X,Y,Z) = (X \text{ Xor } Y)(X \text{ Nor } Y')$

d. $F(X,Y,Z) = (X' + Y' + Z)\,(X + Y)$

۱۴. د هر یو لاند يني تابع (Function) له پاره صدق جدول (Truth Table) لاسته راوړئ.

a. $F(X,Y,Z) = Xy' + Xz' + Yz$

b. $F(X,Y,Z) = (X + Y)(X + Z')$

c. $F(X,Y,Z) = Xy(Y + Z')$

۱۵. لاندې تابع ګانی (Functions) ساده کړئ.

a. $F(X,Y,Z) = Yz + (X + Y)' + (Xyz)'$

b. $F(X,Y,Z) = (X + Y + Z)'(X + Y)$

۱٦. د لاندې تابع ګانو (Functions) له پاره منطق (Logic) سرکټونه رسم کړئ.

a. $F(X,Y,Z) = (X + Y)' + Yz$

b. $F(X,Y,Z) = (Xyz)' + Xz + Yz$

دریم څپرکی

منټرمونه (Minterms) ، مکسټرمونه (Maxterms) ، د کارناف نقشه (Karnaugh Map) ، او عمومي ګیټونه (Universal Gates)

موخې: ددي څپرکي په زده کولو سره به تاسې وکولای شئ چې لاندې کړنې تر سره کړئ :

د بولین تابع (Boolean Function) ښودنه د ضرایبو د مجموعې (Sum Of Product) او د مجموعې ضرب (Product Of Sum) په انځور.

- د تابع له پاره د صدق جدول (Truth Table) جوړول کوم چې د ضرایبو د مجموعې پواسطه ښودل کیږي.
- د تابع له پاره د صدق جدول (Truth Table) جوړول کوم چې د مجموعې ضرب په واسطه ښودل کیږي.
- د جدول څخه په استفادې د تابع جوړول.
- د کارناف نقشې (K- Map) پواسطه د تابع ساده کول.
- د کارناف په نقشه کې د نه پاملرنې (Don’t Care) حالت عملي کول.
- د نانډ (NAND) او نار (NOR) ګیټ په استعمال سره د منطقي سرکټ رسمول.

۱.۳ پېژندنه

ډیجیټل سـرکټ (Digital Circuit) د صـدق جدول یا بولین تابع په واسـطه ښودل کېږي. هغه ډیجیټل سرکټ ده چې زیات ډیجیټل ورکړیز او زیات ډیجیټل راکړیز لري، نو د راکړیز (Outputs) قیمت يي د ورکړیز (Inputs) په اوسـنی قیمت پوری اړه لري.

بولین تابع سـره کولای شـو د مکسټرمونو د ضرب او یا د مینټرمونو د مجموعې په انځور وښـایو تر څو ډیزاینر وکولای شـي په اسـانی سـره تري جدول جوړ کړي. همدارنګه د بولین فرضـیي (Boolean Theorem) د اسـتعمال څخه پرته کولای شو د کارناف نقشې پواسطه د بولین تابع ساده کړو.

زیـاتره ډیجیټـل سـیسـتمونـه د عمومي ګیټونو(Universal Gates) یعنی نـانـډ او نار(NAND , NOR) پواسطه ډیزاین شوی دی.

۲.۳ منټرمونه (Minterms)

منټرمونـه پـه یو تابع کې د متحولونو د هر ترکیب (Combination) سـره یو ځای راتلای شي. که چیرې یو تابع د N شمیر متحولونو لرونکې وي نو د هغې د منترمونو تعداد 2^n سره مساوي دD. فرض کړئ X او Y دوه متحولین دي نو د دې له پاره دلته څلور ممکنـه ترکیـب (Possible Combinations) جوړیـدلای شـي چې عبـارت دي له XY او $X\bar{Y}$ ، $\bar{X}Y$ ، $\overline{XY}$ چې دغه څلورو ممکنه ترکیب ته د X او Y منټرمونه وایي.

۱.۳ جدول منټرمونه او د هغی ډیزاین د $F(X,Y) = (X\ AND\ Y)$ له پاره ښایي.

پـه ۱.۳ جدول کې $\overline{XY = 1}$ که چېرته X= 0 او Y=0 نو $\overline{XY}$ د $m0$ په واسـطه ښـودل کیږي. $\bar{X}Y = 1$ چېرته x= 0 او Y=1 نو $\bar{X}$Y د m1 په واسـطه ښودل کیږي ،$X\bar{Y} = 1$ که چېرته X= 1 او Y=0 نو X$\bar{Y}$ د m2 په واسـطه ښـودل کیږي، XY = 1که چېرته X= 0 او Y=0 نو XY د m1 په واسطه ښودل کیږي.

منتېرم کارېال (**Application Of Minterms**): ددی په واسطه کولی شو چې په اسانی سره صدق جدول (Truth Table) جوړ کړو او هم د صدق جدول (Truth Table) څخه کولی شو منتېرم پیدا کړو. فرض کړئ F(X,Y) = X _Y + X Y تابع او دهغه جدول په ۲.۳ انځور کې ښودل شوی کولی شو دغه تابع د F(X,Y) = m1 + m2 په انځور سره وښايو چې هر منتېرم په جدول کې یو ځرګندوي او هم کولی شوچې د F(X, .Y) = Σ(1,2) په انځور سره وښايو.

درې-متحوله منتېرمونه (**Three-Variable Minterms**): د X، Y او Z متحولونه 8 مینتېرمونه په لاس راکوي لکه څرنګه چې په ۳.۳ جدول کې ښودل شوی دي.

۱.۳ مثال: د لاندې تابع له پاره جدول رسم کړئ.؟

$$F(X,Y,Z) = X'Y'Z + X'YZ + XYZ$$

د F تابع کولی شو چې د منتېرمونو د مجموعې په ډول وښايو (یا هم چېرته چې (F=1) وي.

$$F(X,Y,Z) = m1 + m2 + m3$$

یا

$$F(X,Y,Z) = \sum(1,3,7)$$

X Y	Minterm	Designation
0 0	$\bar{X}\ \bar{Y}$	m_0
0 1	$\bar{X}$ Y	m_1
1 0	X $\bar{Y}$	m_2
1 1	X Y	m_3

۱.۳ جدول د F(X, Y) منتېرم

X	Y	F	
0	0	0	~~m0~~
0	1	1	m_1
1	0	1	m_2
1	1	0	~~m3~~

۲.۳ جدول: د $F(X,Y) = XY' + XY$ تابع له پاره جدول او منټرم

X Y Z	Minterms	Designation
0 0 0	$X'Y'Z'$	m_0
0 0 1	$X'Y'Z$	m_1
0 1 0	$X'YZ'$	m_2
0 1 1	$X'YZ$	m_3
1 0 0	$XY'Z'$	m_4
1 0 1	$XY'Z$	m_5
1 1 0	XYZ'	m_6
1 1 1	XYZ	m_7

۳.۳ جدول: درې – متحوله منټرم

X Y Z	F
0 0 0	0
0 0 1	1
0 1 0	0
0 1 1	1
1 0 0	0
1 0 1	0
1 1 0	0
1 1 1	1

۴.۳ انځور: د $F(X,Y,Z) = X'Y'Z + X'YZ + XY$ تابع له پاره صدق جدول (Truth Table)

په پورته جدول کې په 1، 3او 7 قطار کې 1 سـره مسـاوي دي او نو ټول 0 دي

۴.۳ انځور: سره کولای شو چې یو تابع د منټرم د مجموعې په واسطه هم په لاس راوړو.

۲.۳ مثال لاندې جدول درکړل شوی دی د F تابع یي پیدا کړئ د تابع راکړیز په هغه صورت کې 1 وي کله چې یي ورکړیز 001=M1 ، 011=M3 ، 101=M5 او 111=M7 وي . ددی له پاره $F(X,Y,Z) = m1 + m3 + m5 + m7$ یا هم

$$F(X,Y,Z) = \sum(1,3,5,7)$$

X Y Z	F
0 0 0	0
0 0 1	1
0 1 0	0
0 1 1	1
1 0 0	0
1 0 1	1
1 1 0	0
1 1 1	1

د منټرم هر یو دیزاین د هغې په مربوطه ضریب (′ مثال $m0 = X'Y'Z$) پر ځای وضع کړئ. نو ددی په نتیجه کې لاندې تابع په دی شکل په لاس راځي

$$F(X,Y,Z) = X'Y'Z + X'YZ + XY'Z + XYZ$$

۳.۳ مثال: دلاندې جدول له پاره:

a. د F تابع پیدا کړئ

b. تابع ساده کړئ

د ساده شوی تابع له پاره منطقي سرکټ جوړ کړئ

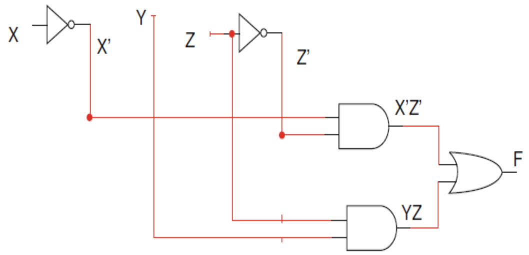
X Y Z	F
0 0 0	1
0 0 1	0
0 1 0	1
0 1 1	1
1 0 0	0
1 0 1	0
1 1 0	0
1 1 1	1

د جدول څخه یوازې هغه قطارونه د منتېرم له پاره انتخابوو چېرته چې تابع $F = 1$ سره وي $m0, m2, m3, m7$

یا

$$F(X,Y,Z) = m0 + m2 + m3 + m7$$
$$F(X,Y,Z) = X'Y'Z' + X'YZ' + X'YZ + XYZ$$
$$F(X,Y,Z) = X'Z'(Y' + Y) + YZ(X' + X)$$
$$F(X,Y,Z) = X'Z' + YZ$$

د ساده شوې تابع F له پاره منطقي سرکټ په ۱.۳ انځور سره ښودل شوی.



۱.۳ انځور: د $F(X,Y,Z) = X'Z' + YZ$ له پاره منطقي سرکټ ښکاره شوی.

۳.۳ مکسټرمونه (**Maxterms**)

مکسټرم (Maxterm) د منټرم (Minterm) کمپلیمنټ (Complement) دی که جیري منټرم (X'Y'Z')M0 وی نو مکسټرم M0 عبارت دي له

$$F(\overline{X'Y'Z'}) = X + Y + Z$$

۵.۳ جدول د دريې متحولونو له پاره مکسټرم څرګندوي

په جدول کې کوم ځای کې چې د تابع راکړیز 1 وي هغه منټرم ښکاره کوي او کوم ځای کې چې راکړیز 0 وي هغه مکسټرم ښکاره کوي

فرض کړئ ۳.٦ صدق جدول (Truth Table) کې چېرته چې د f تابع د مجموعې د ضرب (Product Of Sum) پواسطه ښودل کیږي (نوټ : د مجموعې ضرب د د ضرب د جمعې معکوس دی)

$$F(X, Y, Z) = M0\ M2\ M4\ M5\ M6$$

او یا په لاندې شکل هم ښودل کیږي

$$F(X, Y, Z) = \pi(0, 2, 4, 5, 6)$$

مکسټرم هر ډیزاین سمبول د هغې مربوطه حاصل سره تعویض کوو چې په نتیجه کې

$$F(X, Y, Z) = (X + Y + Z)(X + Y' + Z)(X' + Y + Z)(X' + Y + Z')$$

X Y Z	Maxterm	Designation
0 0 0	X + Y + Z	M_0
0 0 1	X + Y + Z′	M_1
0 1 0	X + Y′ + Z	M_2
0 1 1	X + Y′ + Z′	M_3
1 0 0	X′ + Y + Z	M_4
1 0 1	X′ + Y + Z′	M_5
1 1 0	X′ + Y′ + Z	M_6
1 1 1	X′ + Y′ + Z′	M_7

۵.۳ جدول د F(X, Y, Z) تابع مکسټرم

X Y Z	F
0 0 0	0
0 0 1	1
0 1 0	0
0 1 1	1
1 0 0	0
1 0 1	0
1 1 0	0
1 1 1	1

۶.۳ جدول د F(X,Y,Z) = M0 M2 M4 M5 M6 تابع له پاره صدق جدول (Truth Table)

۴.۳ کارناف نقشه (K-Map)

کارناف نقشه ددې له پاره استعمالېږي چې یو تابع ساده کړو غیر له بولین الجبرا د استعمال څخه. همدارنګه د کارناف نقشه د صدق جدول (Truth Table) د جوړېلو له پاره یوه بله لاره ده . د کارناف نقشــه د ډېرو خانو (Cells) څخه جوړه شــوې ده چې هره خـانه يي يو منتېرم ښرګندوي. هغه خانې چې د 1 په واســطه ډکې شـــوي دي نو منتېرم دا ښرګندوي چې د هغی د منتېرمونو جمع نیول کیږي تر څو یو تابع وښایئ او بر عکس هغه خاني چې د صفر پواسطه ډکې شوې دي د هغې د مکسټرمونو ضرب ښرګندوي.

د X او Y دوه متحولین څلور منتېرمونه ښـکاره کوي لکه په ۷.۳ جدول کې ښـودل شـوي

دي هر منټرم د کارناف په نقشه کې د یوې خانې پواسطه ښودل کیږي نو په همدې اساس د دوه متحولینو لرونکې کارناف نقشه څلور خانې لري لکه په ۲.۳ انځور: کې ښودل شوي دي.

په ۲.۳ انځور: کې هره خانه یو منټرم ښکاره کوي هغه خانه چې په صفر سطر او ستون کې موقعیت لري M0(Minterm Zero) یا $X'Y'$ ښکاره کوي او هغه خانه چې په 1 سطر او 1 ستون کې موقعیت لري m3 یا XY ښکاره کوي لکه څنګه چې په ۲.۳ انځور: کې ښودل شوي د دواړه خانو په صفر سطر کې X' موجود دی نو پدې اساس د X' سطر پواسطه په نښه شوی ده، دواړه خانې په لومړی سطر کې د X لرونکې ده نو څکه دغه سطر د X پواسطه په نښه شوی دی.

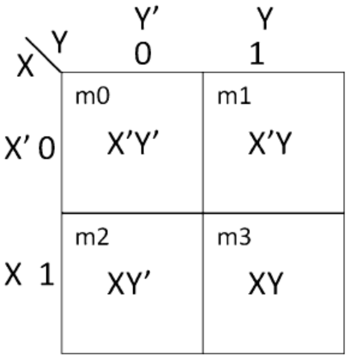
د یوی تابع د کارناف نقشه دصدق جدول (Truth Table) د جوړولو له پاره یو بله لار ده چې په ۳.۳ انځور: کې ښودل شوې ده

د $F(X,Y) = XY' + X'Y' = m0 + m1$ تابع فرض کړئ. د نوموړې تابع له پاره په ۸.۳ انځور: کې جدول (Truth Table) ورکړل شوی دی او تابع کارناف نقشې ته هم تبدیله شوې ده لکه څرنګه چې په ۳.۳ انځور: کې ښودل شوې ده.

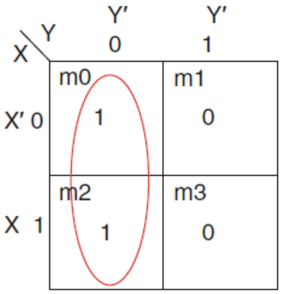
ګاونډې خانې (Adjacent Cells): دوه خانې په هغه وخت کې ګاونډې وي کله چې یوازې په یو متحول کې د یو بل سره فرق ولري د X'Y' او X'Y خانې د یو بل ګاونډئ خانې دي څکه چې دلته یوازې په Y او Y' کې سره فرق لري. ګاونډې خانې کولی شو چې سره یو ځای کړو ددې له پاره چې تر څو د کارناف نقشه ساده او اسانه شي.

X Y	Minterms	Designation
0 0	X′Y′	m_0
0 1	X′Y	m_1
1 0	XY′	m_2
1 1	XY	m_3

۷.۳ جدول د دوه متحولینو له پاره منټرم



۲.۳ انځور: د دوه متحول لرونکي کارناف نقشه



۳.۳ انځور: په کارناف نقشه کې تابع

X Y	F
0 0	1
0 1	0
1 0	1
1 1	0

۸.۳ جدول د F(X,Y) = XY' + X'Y' تابع له پاره د صدق جدول (Truth Table) په کارناف نقشه کې ښودل کیږي چې د $m0$ او $m2$ خانې د 1 درلودونکي دي او پاتې یې صفر دی او همدارنګه د $m0$ او m2 خانې د یو بل ګاونډی خانې (Adjacent Cells) دي .

نوټ: ګاونډی خانی د ′Y ټول ستون (Column) راخلي او نور ټول صفر دي نو ستاسې ساده شوې تابع عبارت ده له $F(X,Y) = Y'$.

۴.۳ مثال: لاندې تابع ساده کړئ:

$$F(X,Y) = X'Y + XY' + XY$$

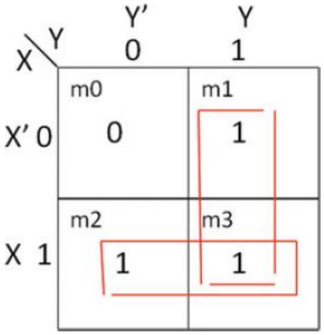
یا

$$F(X,Y) = m1 + m2 + m3$$

منټرم د کارناف نقشې ته استول نتیجه یې په ۴.۳ انځور کې. په ۴.۳ انځور کې ښودل شوي چې m2 او m3 خانی د یو بل ګاونډي خانې دي نو پدې اساس کولای شو چې دغه دواړه خانې د یو بل سره یو ځای کړو په همدی ترتیب د m1 او m3 خانی هم یو ځای کوو، د کارناف نقشې په لوستلو سره کولای شو چې ساده شوی تابع په لاس راوړو .

د m2 او m3 د ټول سطر له پاره X او د m1 او m3 د ټول ستون له پاره Y او نور خانې ټولې صفر دي نو ددی له پاره،

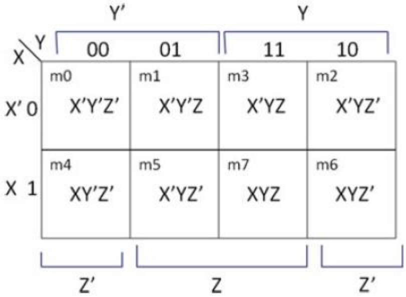
$$F(X,Y) = X + Y$$



۴.۳ انځور: د F(X,Y) = X'Y + XY' + XY تابع له پاره د کارناف نقشه

۱.۴.۳ درې – متحوله نقشه (Three-Variable Map): درې – متحوله

نقشه 8 خانی لري چې هره خانه یې منټرم ښکاره کوي لکه څرنګه چې په ۵.۳ انځور کې ښودل شوي دي.



۵.۳ انځور: درې – متحوله نقشه

د ۵.۳ انځور د کارناف نقشې له پاره بايد لاندې نقطې په ياد ولرئ

a) په 0 سطر کې ، ټولې خانې د X' درولودنکې کې دي نو ددې په اساس نوموړي سطر د X' پواسطه په نښه شوی

b) په 1 سطر کې ، ټولې خانې د X درولودنکې دي نو ددې په اساس نوموړی سطر د X پواسطه په نښه شوی

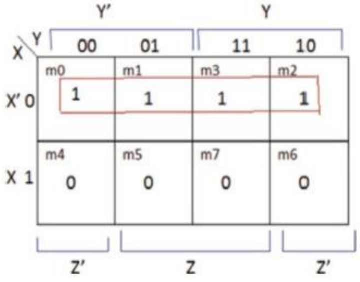
c) په 11 او 01 ستون کې ، ټولې خانې د Y درلودنکې دي نو ددې په اساس نوموړی سطر د Y پواسطه په نښه شوی

d) په 00 او 01 ستون کې ، ټولې خانی د Y' درولودنکې دي نو ددې په اساس نوموړی سطر د Y' پواسطه په نښه شوی

e) په 01 او 11 ستون کې ، ټولې خانی د Z درولودنکې دي نو ددې په اساس نوموړی سطر د Z پواسطه په نښه شوی

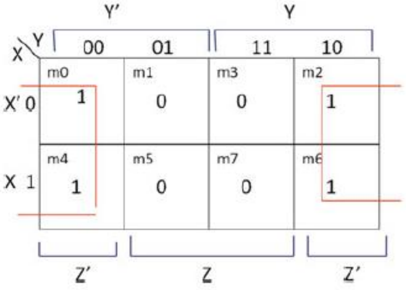
f) په 00 او 10 ستون کې ، ټولې خانی د Z' درولودنکې دي نو ددې په اساس نوموړی سطر د Z' پواسطه په نښه شوی.

ګاونډې خانې په کارناف نقشه کې د يو بل سره ګروپ کيږي د کارناف په نقشه کې 2 خانې ، 4 خانې ،8 خانې ، 16 خانې يوځای کېدای شي. ۶.۳ انځور څرګندوي چې څنګه کولی شو چې په کارناف نقشه کې د 1 خانې سره ګروپ کړو.

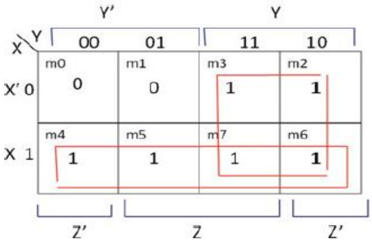


۶.۳ انځور د کارناف نقشې د څلورو خانو ګروپ کول چېرته چې ساده شوی تابع عبارت ده له

$$F(X, Y, Z) = Z'$$



۷.۳ انځور د کارناف په نقشه کې څلورو Z' خانو ګروپ کول



۸.۳ انځور د ورکړل شوی تابع له پاره د کارناف نقشه کې د خانو یو ځای کول.

۵.۳ مثال: لاندې تابع ساده کړئ؟

F (X;Y; Z) = X'Y + XZ + XZ

لومړی هره یوه برخه باید د کارناف نقشې ته ولیږل شي

(a) نو لومړی برخه X'Y ده نو د X' سطر او Y ستون په تقاطع کې 1 ږدو لکه څرنګه چې په ۸.۳ انځور کې ښودل شوی (m3,m2)

(b) دوهمه برخه د XZ ده نو د Y' ستون او X سطر په تقاطع کې 1 ږدو (m5,m7)

(c) دریمه برخه يي د XZ′ ده نود X سطر او Z ستون په تقاطع کې 1 ږدو (m4 , m6)

تر اوسه پورې ګاونډي خانې د X سطر او د Y خانې په شمولیت سره شاملې کړي چې د نوموړي تابع ساده شکل عبارت دی له $F(X,Y,Z) = X + Y$

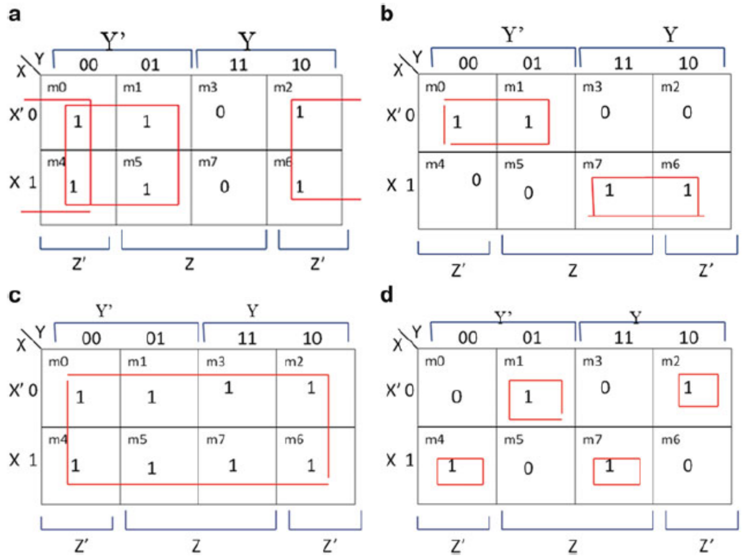
٦.۳ مثال: د ۹.۳ انځور کارناف نقشه د a-d پورې ولولئ او ساده شوی تابع يي معلومه کړئ

a) ټول ګاونډی خانی په Z' او Y ستون کې د 1 درلودونکې دي

F (X;Y; Z) = Z+ Y

b) د X' سطر او Y' ستون خانه د X سطر او Y ستون د خانی سره ګاونډی ده

$$F(X,Y,Z) = X'Y' + XY$$



۹.۳ انځور کارناف نقشه د ٦.٣ مثال له پاره

c) ټولې خانې د 1 قیمت درلودونکي دی نو پدې اساس تابع د 1 سره مساوي ده

$$F(X,Y,Z) = 1$$

d) د ګاونډی خانې په نظر کې نه نیولو په نتیجه کې ساده شوې تابع عبارت ده له

$$F(X,Y,Z) = X'Y'Z + X'YZ' + XY'Z' + XYZ$$

۲.۴.۳ څلور – متحوله نقشه (**Four-Variable K-Map**): څلور – متحوله نقشه د 16 خانو درلودونکې وی لکه په ۱۰.۳ انځور کې ښودل شوی دی مهرباني وکړئ د نقشې خاص جوړښت نوټ کړئ

د هر متحول له پاره د کارناف نقشه کې د ځای انتخاب په لاندې ډول دی:

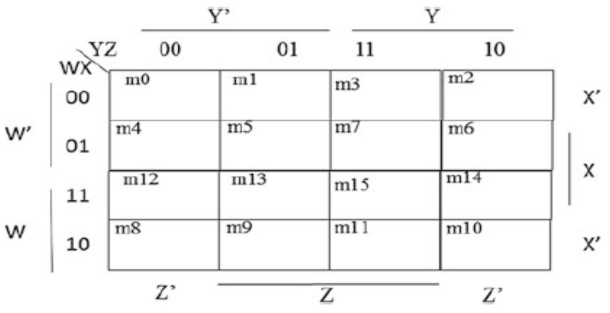
- W په 11 او 10 سطر (Row) کې راځي
- W' په 00 او 01 سطر کې راځي

- X په 01او 11 سطر کې راځي
- X’ په 00او 10 سطر کې راځي
- Y په 11او 10 ستون کې راځي
- Y’ په 00او 01 ستون کې راځي
- Z په 01او 11 ستون کې راځي
- Z’ په 00او 10 ستون کې راځي

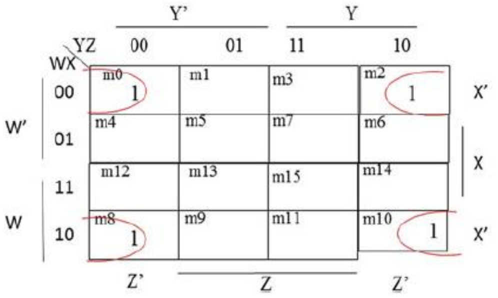
۷.۳ مثال: لاندې تابع ساده کړئ ؟

$$F(W,X,Y,Z) = m0 + m2 + m8 + m10$$

تابع په لاندې ډول د کارناف نقشې ته لیږو لکه په **۱۱.۳** انځور کې ښودل شوي دي. که چېرې د کارناف نقشه د منځ څخه په افقی او عمودی ډول پراخه شی نو ټولې څلور خانې يي:



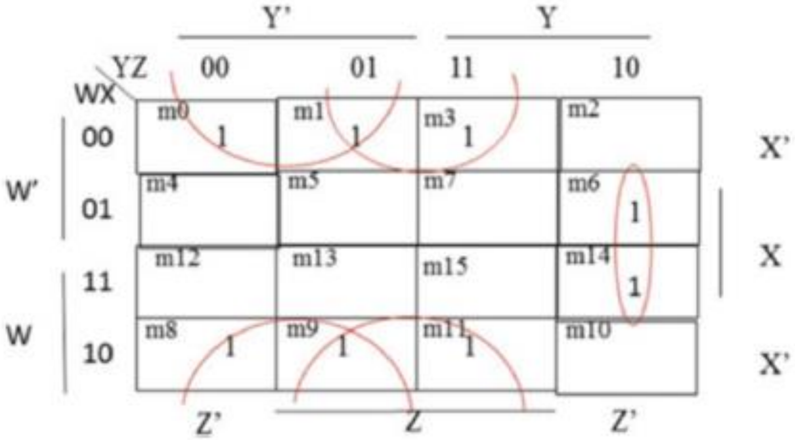
۱۰.۳ انځور: څلورمتحوله نقشه



۱۱.۳ انځور: د $F(W,X,Y,Z) = m0 + m2 + m8 + m10$ تابع له پاره د کارناف نقشه

د 1 قیمت درلودونکي دي. نوټ د دغه څلورو خانو د تقاطع څخه X’ او Z’ په لاس راځي نو په نتیجه کې ساده شوی تابع یې عبارت ده له:

$$F(W,X,Y,Z) = X'Z'$$



۸.۳ مثال: لاندې د کارناف نقشه ولولئ:

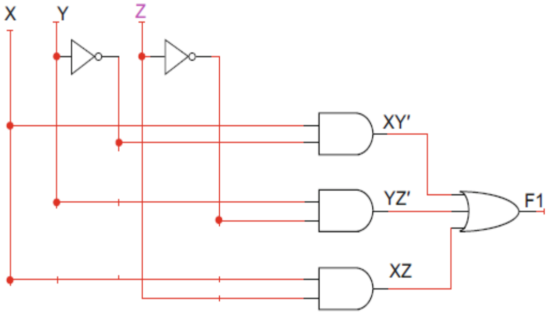
$$F(W,X,Y,Z) = X'Y' + X'Z + XYZ'$$

۵.۳ د ضرایبو مجموعه (SOP) او د مجموعې ضرب (POS)

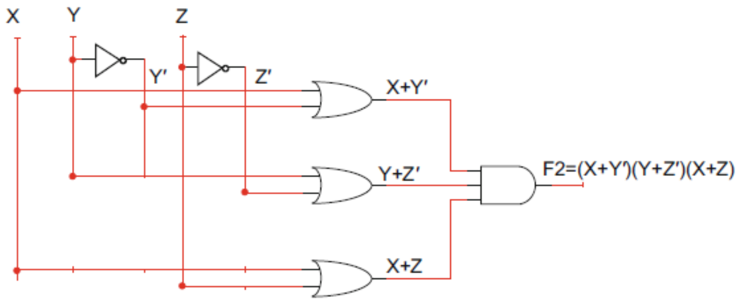
Sum Of Products (Sop) And Product Of Sums (POS))

د یوی تابع د ضـرایبو مجموعه (Sum Of Product) عبارت ده له ساده شوي منترمونو د مجموعې (Sum Of Minterms) څخه.

د $F(X,Y) = XY' + XY$ تابع ته وګورئ کوم چې د ضرایبو د مجموعې (SOP) په ډول ده چې د جمعي علامه یي د ار (OR) معنا لري او XY ته ضـرب ویل کیږي. فرض کړئ د $F1(X,Y,Z) = XY' + YZ' + XZ$ تـابع او ۱۲.۳ انځور د F1 تابع له پاره منطقي سـرکټ ښـکاره کوي کوم چې د انډ – ار (AND – OR) څخه جوړ شـوی دی.



۱۲.۳ انځور د $F1(X,Y,Z) = XY' + YZ' + XZ$ تابع له پاره منطقي سرکټ کوم چې د انډ او ار (AND – OR) څخه جوړ شوی دی



۱۳.۳ انځور د $F2(X,Y,Z) = (X + Y')(Y + Z')(X + Z)$ تابع له پاره منطقي سرکټ.

فرض کړئ د $F2(X,Y,Z) = (X + Y')(Y + Z')(X + Z)$ تـابع کوم چې د مجموعې ضرب (Product Of Sum) په انځور ښودل شوی دی او ۱۳.۳ انځور د F2 تابع له پاره منطقي سرکټ ښکاره کوي کوم چې د ار – انډ (OR – AND) څخه جوړ شوی دی

۹.۳ مثال: لاندې تابع د ضرایبو د مجموعې (SOP) او د مجموعې د ضرب (POS) په شکل ساده کړئ.

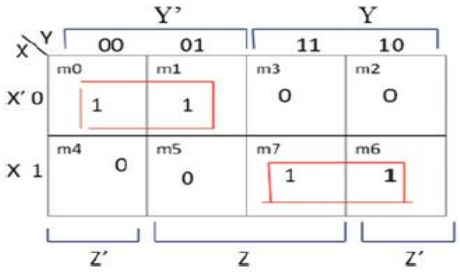
$$F(X,Y,Z) = \sum(0,1,6,7)$$

په کـوم ځای کې 1 راغـلی وي هغه له یو بـل سـره یـو ځـای کوو (m0 ,m1 , m6 , m7) لکه په ۱۴.۳ انځور کې ښودل شوی دی.
د ضرایبو مجموعه (Sum Of Product) یې مساوي کیږي په:

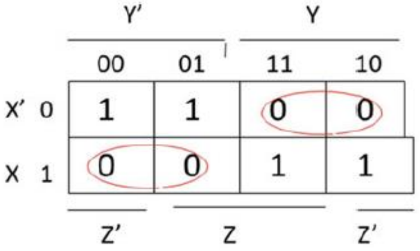
$$F(X,Y,Z) = X'Y' + XY$$

د کارناف په نقشـه کې د 0 په یو ځای کولو سره یو مهم مکسټرم په لاس راځي لکـه څرنګـه چې پـه ۱۵.۳ انځور کې ښـودل شـوی دی که چیرې F مسـاوي شـي له $\sum(0,1,6,7)$ نو F′ مساوي کیږي له $\pi(M2, M3, M4, M5)$ سره.
تر اوسـه د مکسـټرمونو ضـرب (Product Of Maxterms) مسـاوي دی د F د کمپلیمنټ (Complement) سـره. نو ددې لـه پـاره چې F پیـدا کړو نو د تـابع دواړه خواوی کمپلیمنټ نیسو.

$$F(X,Y,Z) = (XY' + X'Y)'$$



۱۴.۳ انځور د $F(X,Y,Z) = \sum(0,1,6,7)$ تابع له پاره د کارناف نقشه



۱۵.۳ انځور د $F'(X,Y,Z) = \pi(M2, M3, M4, M5)$ تابع له پاره د کارناف نقشه

د ديمورګان فرضیې (Demorgan's Theorem) څخه په استفادې:

$$F(X,Y,Z) = (XY')'(X'Y)'$$

یا:

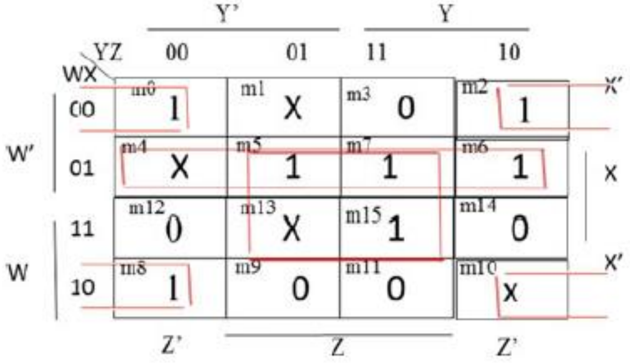
$$F(X,Y,Z) = (X' + Y)(X + Y')$$ د مجموعې د ضرب په شکل.

۶.۳ د نه پاملرنې حالتونه (Don't Care Conditions)

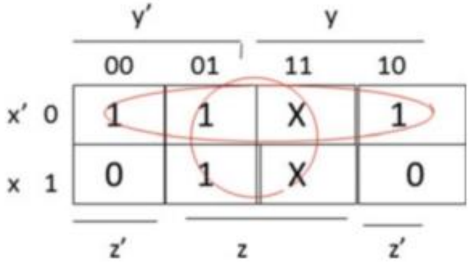
که چېرته په یو صــدق جدول (Truth Table) کې د ورکړیزو متحولونو یو ځای کېدل ناممکن وي نو دغه حالت ته د نه پاملرنې حالتونه ویل کیږي. په دغه حالتونو کې د یوی تـابع راکړیز قیمت د اهمیت وړ ندی د مثال په ډول د لســو په قاعده دوه –یزه کوډ (BCD) چې پدې کې اصــلا څلور بیټونه دي چې یوازې د 0000 څخه تر 1001

پورې استفاده کوي، فلهذا د 1010 څخه تر 1111 پورې BCD نه ده، نو د جدول او يا هم د کارناف نقشې قيمتونه د نه پاملرنې حالتونه دي د جدول يا کارناف نقشې خانې د X يا D پواسطه په نښه کيږي چې دا د نه پاملرنې حالت معنا ورکوي او ددی راکړيز قيمت که 1 وی يا 0 وي هيڅ تاثير نه غورځوي. د نه پاملرنې حالت ددې په خاطر استعماليږي چې په کارناف نقشه کې د خانو تر منځ ګاونډيتوب پراخه کړي تر څو يوه ساده تابع په لاس راشي. تر اوسه پورې د نه پاملرنې راکړيز قيمت د اهميت وړ ندی.

۱۰.۳ مثال: په ۱٦.۳ انځور کې يو کارناف نقشه د نه پاملرنې منترم سره ښودل شوې ده په m1 , m10 , m13 کې. تر اوسه د نه پاملرنې حالت قيمت به يا يو وي يا هم صفر وي. مونږ د خانو تر منځ د ګاونډيتوب د پراخولو له پاره د خانی قيمت 1 فرضوو تر څو په اسانئ سره يو ځای شي.



۱٦.۳ انځور يو کارناف نقشه د نه پاملرنې منترم سره



۱۷.۳ انځور د F(X,Y,Z) = M0 +M1+M2+M5 او D(X,Y,Z) = M3 + M5 توابعو له پاره د کارناف نقشه

له ۱۶.۳ انځور په بنیاد تابع $F(W,X,Y,Z) = XZ + X'Z' + XW'$ وي په هغه صورت کې چې د F تابع له پاره منټرم د نه پاملرنې حالت وي.
د نه پاملرنې تابع D مساوي ده د نه پاملرنې د منټرمونو د مجموعې سره. که چیرې m7 یوازینی د نه پاملرنې حالت وي نو د نه پاملرنې تابع یې د D(X,Y,Z) = m7 پواسطه ښکاره کیږي.

۱۱.۳ مثال: لاندې تابع ساده کړئ په داسی حال کې چې D د نه پاملرنې تابع ده

$$F(X,Y,Z) = m0 + m1 + m2 + m5$$

$$D(X,Y,Z) = m3 + m7$$

ددغو قیمتونو نتیجه د کارناف نقشه کې ولیکئ ۱۷.۳ انځور په اساس د ګاونډیو خانو د ګروپ کولو او د نه پاملرنې د حالت څخه په استفادې سره:

F(X,Y,Z) = X' + Z

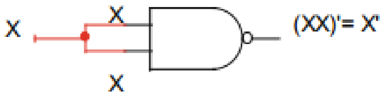
۷.۳ عمومي ګیټونه (Universal Gates)

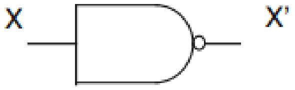
د ناند (NAND) او نار (NOR) ګیټونو ته عمومي ګیټونه ویل کیږي. د ناند او نار ګیټونو پواسطه یو ډیزاینر کولی شي چې نور منطقي ګیټونه جوړ کړي لکه ار (OR)، انډ (AND) او ناټ (NOT) ګیټونه.

۱.۷.۳ د ناند (NAND) ګیټ استعمال:

a) ناند (NAND) ګیټ څخه د ناټ (NOT) ګیټ جوړول:

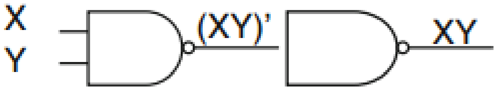
ناټ ګیت د ناند ګیټ د ورکړیزو قیمتونو د یو ځای کېدو څخه جوړیږي لکه څرنګه چې په لاندې شکل کې ښودل کیږي.
یا هم پدې انځور یې هم ښودلی شو:





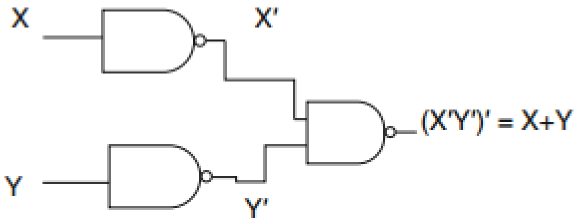
b) نانډ (NAND) ګیټ څخه د انډ (AND) ګیټ جوړول:

انډ ګیټ د نانډ ګیټ د ورکړیزو ورکول یو نانډ ګیټ ته او د همدي ګیټ د راکړیز قیمت له پاره یو بل نانډ ګیټ جوړول لکه په لاندي شکل کې ښودل شوي:



c) نانډ (NAND) ګیټ څخه د ار(OR) ګیټ جوړول:

د ار ګیټ د جوړولو له پاره د هر ورکړیز قیمت له پاره جلا جلا یو نانډ ګیټ جوړوو چې په نتیجه کې د هر نانډ ګیټ د راکړیز قیمت له پاره یو نانډ ګیټ جوړو لکه په لاندي انځور کې ښودل شوي دي.



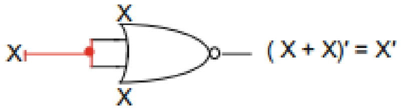
۲.۷.۳ د نار (NOR) ګیټ استعمال:

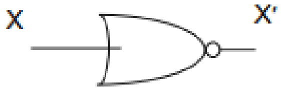
a) د نار(NOR) ګیټ څخه د ناټ (NOT) ګیټ جوړول:

ناټ ګیت د نار ګیټ د ورکړیزو قیمتونو د یو ځای کېدو څخه جوړیږي لکه څرنګه چې

په لاندې انځور کې ښودل کیږي

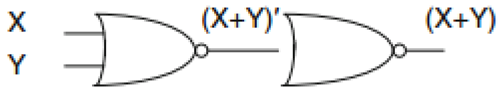
او یا هم پدې انځور یې هم ښولای شو:





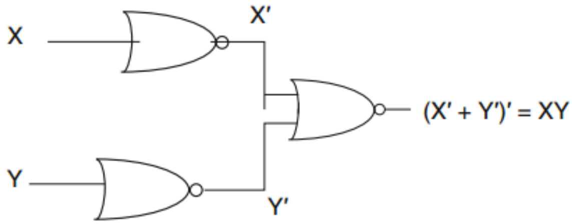
b) نار (NOR) ګیټ څخه د ار (OR) ګیټ جوړول:

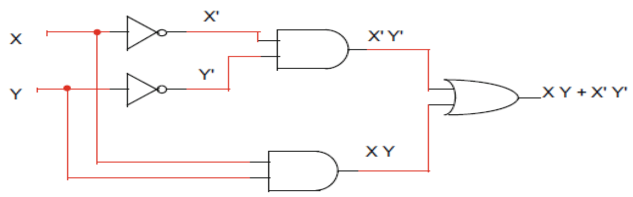
ار (OR) ګیټ د نار (NOR) ګیټ د ورکړیزو ورکول یو نار ګیټ ته او د همدی ګیټ د راکړیز قیمت له پاره یو بل نار ګیټ جوړول لکه په لاندې انځور کې چې ښودل شوی.



c) نار(NOR) ګیټ څخه د انډ (AND) ګیټ جوړول:

د انډ (AND) ګیټ د جوړولو له پاره د هر ورکړیز قیمت له پاره جلا جلا یو نار ګیټ جوړوو چې په نتیجه کې د هر نار ګیټ د راکړیز قیمت له پاره یو نار ګیټ جوړو لکه په لاندې انځور کې ښودل شوي دي:





۱۸.۳ انځور د F(X, Y) = XY + X'Z' تابع له پاره منطقي سرکټ

۳.۷.۳ د منطقي تابع ګانو یوازې تطبیق کول د نانډ (NAND) یا نار (NOR) ګیټونو په کارولو سره (Implementation Of Logic Functions Using NAND Gates Or NOR Gates Only):

ددې پرځای چې د مختلفو ګیټونو د ترکیب کارولو سره منطقي تابع ګانې تطبیق شي بلکې زیاتره منطقي تابع ګانې یوازې د نانډ (NAND) یا نار (NOR) ګیټونو په کارولو سره تطبیق کیږي ځیرې د منطقي ګیټ مدغم شوي سرکټونه (ICs) یو ډول ګیټونه لري لکه مدغم شوی سرکټ (IC) چې اته انډ (AND) ګیټونه لري. د یو ډول ګیټونو استعمال کېدلی شي د اړینو مدغم شوو سرکټونو شمیر کم کړي. $F(X,Y) = X'Y' + XY$ تابع او ددی منطقي سرکټ دیاګرام په ۱۸.۳ انځور کې په پام کې ونیسئ، نوموړی دیاګرام د انډ (AND) ګیټونو له پاره یو مدغم شوی سرکټ د ناټ (NOT) ګیټونو له پاره او یو مدغم شوی سرکټ د ار (OR) ګیټونو له پاره چې په ټوله کې درې جلا جلا مدغم شوي سرکټونو ته ضرورت لري.

د نوموړی تابع د دوه ځله کمپلیمنټ (Complement) په اخیستلو سره د معادلی ښیي اړخ د نار (NOR) او نانډ (NAND) ګیټونو سره کارول اسانه کېدای شي.

۱۲.۳ مثال: د $F(X,Y) = X'Y' + XY$ تابع له پاره یو منطقي سرکټ جوړکړئ او یوازی نانډ او یوازی نار ګیټ استعمال کړئ

۴.۷.۳ د نانډ (NAND) ګیټ استعمال:

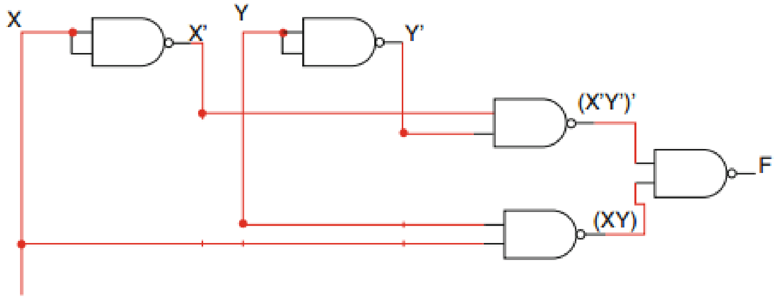
- د مساوات د دواړو خواو کمپلیمنټ (Complement) نیسو
- $F(X,Y) = X'Y' + XY \rightarrow F(X,Y) = [(X'Y' + XY)']'$
- د بولین فرضیي په استعمال سره کولی شو نانډ د پوهیدو وړ شی

$$F(X,Y) = [(X'Y')' + (XY)']'$$

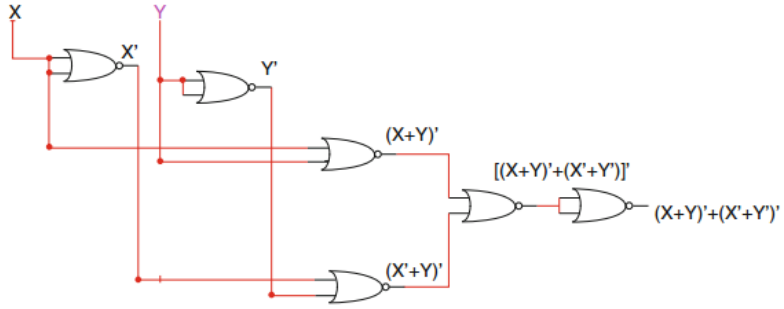
په اخره کې لاسته راغلی تابع یو ځل بیا وګورئ $F(X,Y) = [(X'Y')' + (XY)']'$ او د داخلی برخې سره یي تعویض کړئ. ۱۹.۳ انځور

- $F(X,Y) = [(X'Y')' + (XY)']' = [AB]' \ (A \ NAND \ B)$
  - $A = (X'Y')' = X' NAND \ Y'$
  - $B = (XY)' = X \ NAND \ Y$
  - $X' = X \ NAND \ X$

$$Y' = Y \ NAND \ Y$$



۱۹.۳ انځور د $F(X,Y) = X'Y' + XY$ تابع له پاره منطقي سرکټ یوازې د نانډ ګیټ څخه په استفادې.



۲۰.۳ انځور د $F(X,Y) = X'Y' + XY$ تابع له پاره منطقي سرکټ یوازې د نار ګیټ څخه په استفادې.

۵.۷.۳ د نار(NOR) ګیټ استعمال:

- د مساوات دواړه خواوو دوه ځلې کمپلمنټ نیسو
  - $F(X,Y) = X'Y' + XY \rightarrow F(X,Y) = [(X'Y' + XY)']'$
- د بولین فرضیې په استعمال سره کولی شو نار د پوهیدو وړ شي
  - $F(X,Y) = [(X'Y')' + (XY)']'$
  - $F(X,Y) = [(X + Y)(X' + Y')]'$
  - $F(X,Y) = [(X + Y)' (X' + Y')']'$

اخرې لاسته راغلې تابع یو ځل بیا وګورئ $F(X,Y) = [(X + Y)' (X' + Y')']'$ او دداخلي برخې سره یې تعویض کړئ (انځور.۳.۲۰)

- $F(X,Y) = [(X'Y')'(XY)'] = [A + B]' (A\ NOR\ B)$
  - $A = (X + Y)' = X\ NOR\ Y$
  - $B = (X' + Y') = X'\ NOR\ Y'$
  - $X' = X\ NOR\ X$
  - $Y' = Y\ NOR\ Y$

۸.۳ خلاصه

- ډیجټل سرکټ د مختلفو ګیټونو د یو ځای کېدو څخه چې زیات ډیجیټل ورکړېز او راکړېز معلومـات لري جوړ شـــوی دی . راکړېز معلومـات یي د ورکړېز په اوســـنیو معلوماتو پوری اړه لري.
- ترکیبي منطقي سرکټ کولای شو د بولین تابع او یا د جدول پواسطه وښایو.
- بولین فرضیه او د کارناف نقشه د بولین تابع د ساده کولو له پاره استعمالېږي.
- بولین تـابع د ضـــرایبو د مجموعې (Sum Of Product) او یـا هم د مجموعي ضرب(Product Of Sum) پواسطه ښودلای شو
- د نانډ (NAND) او نار (NOR) ګیټ ته عمومی ګیټونه (Universal Gates) ویل کیږي چې د همدی ګیټونه په واسطه نور ګیټونه جوړیږي
- د نه پاملرنې حالت (Don’t Care Condition) هغه ورکړېز معلومات دي چې پـه ترکیبي ســـرکټ کې نه اســـتعمالېږي او په نتیجه کې یي راکړېز معلومات یو د نه پاملرنې حالت (1 , 0) وي
- د ۱ ، ۲ او ۳ څپرکو کې مو د اسـاسي موضوعاتو په اړه بحث وکړ ترڅو وکولای شو یو ډیجیټل سـرکټ ډیزاین کړو . په ۴ څپرکی کې به ولولو چې څنګه کولای شـو یو ډیجیټل سـیستم ډیزاین کړو او هم به د ډیجیټل سیستم د ډیزاین له پاره مهمې برخې وپیژنو د مثال په ډول اصلي بڼې ته تبدیلونکی (Decoder)، ملټپلیکسـر (Multi-Plexer)، دوه یـز جـمـع کـوونـکی(Binary Adder)، دوه یز منفي کوونکی (Binary Subtractor)، حسابي منطقي واحد (ALU) .

ستونزې

۱. د صدق جدول(Truth Table) له پاره راکړیزه تابع پیدا کړئ

a) د منټرمونو د مجموعې په ډول

b) د مکسټرمونو د ضرب په ډول

X Y Z	F
0 0 0	1
0 0 1	0
0 1 0	0
0 1 1	1
1 0 0	0
1 0 1	1
1 1 0	0
1 1 1	1

A B C D	F
0 0 0 0	1
0 0 0 1	0
0 0 1 0	1
0 0 1 1	1
0 1 0 0	0
0 1 0 1	1
0 1 1 0	1
0 1 1 1	1
1 0 0 0	1
1 0 0 1	1
1 0 1 0	0
1 0 1 1	0
1 1 0 0	0
1 1 0 1	1
1 1 1 0	1
1 1 1 1	1

۲. د لاندې توابعو له پاره صدق جدول (Truth Table) جوړ کړئ؟

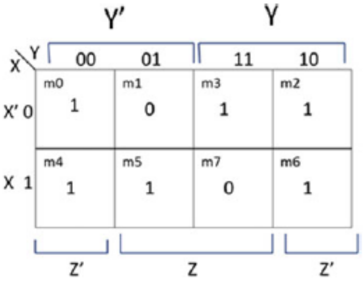
(a $F(X,Y,Z) = \sum(1,3,6,7)$

(b $F(X,Y,Z) = \pi(1,3,4)$

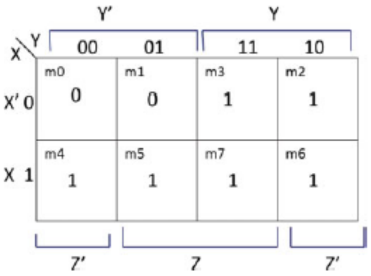
(c $F(W,X,Y,Z) = \sum(1,4,7,10,12,15)$

(d $\mathrm{F(W,X,Y,Z)} = \pi(2,3,4,7,10,11,12,13)$

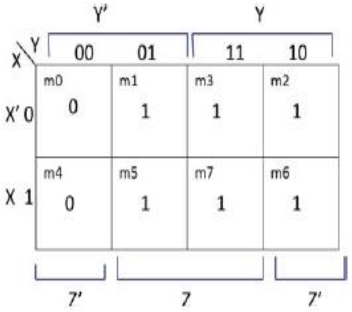
۳. د لاندې کارناف نقشو له پاره د F تابع ولیکئ:



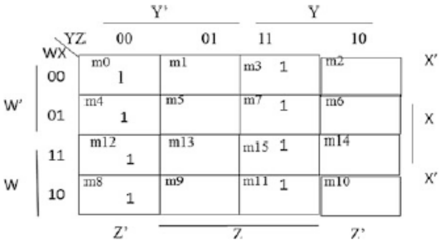
(a



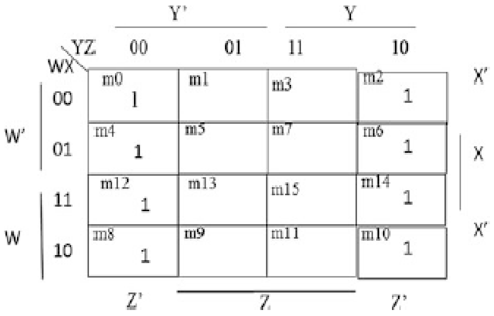
(b



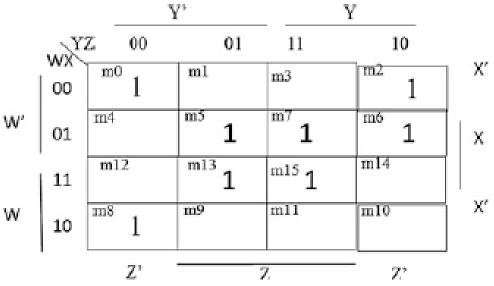
(c



(d



(e



(f

۴. د کارناف نقشې خخه په استفادی د لاندې جدول له پاره ساده راکړیزه تابع پیدا کړئ.

X	Y	F
0	0	1
0	1	1
1	0	1
1	1	0

(a

(b

X	Y	Z	F
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

(c

A B C D	F
0 0 0 0	1
0 0 0 1	0
0 0 1 0	1
0 0 1 1	1
0 1 0 0	0
0 1 0 1	1

(d

A B C D	F
0 1 1 0	1
0 1 1 1	1
1 0 0 0	1
1 0 0 1	1
1 0 1 0	0
1 0 1 1	0
1 1 0 0	0
1 1 0 1	1
1 1 1 0	1
1 1 1 1	1

۵. لاندې تابع ګانې د کارناف نقشې په استعمال سره ساده کړئ ؟

a. $F(X,Y) = m2 + m3$

b. $F(X,Y) = X + X'Y$

c. $F(X,Y) = X' + XY'$

d. $F(X,Y,Z) = m0 + m2 + m5 + m7$

e. $F(X,Y,Z) = X'Y'Z' + X'YZ + XY'Z + XYZ$

f. $F(X,Y,Z) = \pi(0,2,5,7)$

g. $F(X,Y,Z) = XY'Z + X' + Z + Y'Z'$

.h $F(W,X,Y,Z) = X'Y'Z' + XYZ' + WXY + W'X'Y' + WZ$

.i $F(W,X,Y,Z) = X' + XZ' + WX'Y + W'Y' + WZ$

۶. لاندې تابعګانی ساده کړئ چېرته چې D د نه پاملرنې تابع ده ؟

.a F(X,Y,Z) = ∑(0,2,5,7)

$$D(X,Y,Z) = \sum(2,6)$$

.b F(W,X,Y,Z) = ∑(0,1,3,5,9,11)

$$D(W,X,Y,Z) = \sum(2,4,8,10)$$

۷. لاندې تابعګانی د SOP او POS په انځور باندی ساده کړئ او منطقي سرکټ يي رسم کړئ ؟

.a F(X,Y,Z) = ∑(0,2,5,7)

.b F(W,X,Y,Z) = ∑(0,1,4,6,9,11,13,15)

۸. د ٦ مثال د ساده شوی تابع له پاره منطقي سرکټ رسم کړئ

.a نانډ (NAND) ګیټ په استعمال سره

.b نار (NOR) ګیټ په استعمال سره

۹. لاندې تابع ساده کړئ او منطقي سرکټ يي رسم کړئ د

.a نانډ (NAND) ګیټ په استعمال سره

.b د نار (NOR) ګیټ په استعمال سره

F(W,X,Y,Z)=W^’X^’Z^’ + XY^’Z^’+WX+WY

څلورم څپرکی

ترکیبي منطق (Combinational Logic)

موخې: ددغه څپرکي په پای کې، تاسو باید وکولې شې چې:

- د ورکړل شوی دیجیټل سرکټ ورکړیز (Output) پیدا کړئ.
- د مشکلاتو تشریح څخه په استفادې د ترکیبي منطق (Combinational Logic) سرکټ دیزاین کړئ.
- اصـــلي بڼې تـه بـدلوونکې (Decoder) عملي او کـاریـال (Application) زده کړئ.
- بل حالت ته بدلوونکې (Encoder) کاریال (Application) زده کړئ.
- د ملټیپلیکسر (Multiplexer) دندی زده او دیزاین کړئ.
- د مـنطقي (Logic) ګیټونو (Gates) پـه مرســتـه نیمـه جمع کوونکی (Half Adder)، پوره جمع کوونکی (Full Adder) تاسیس (Develop) کړئ.
- د پوره جمع کوونکی (Full Adder) څخه په اســتفادې د دوه ییزه جمع کوونکي (Binary Adder) او منفي کوونکي (Subtractor) دیزاین کړئ.
- دا زده کړئ چې څنګـه ریـاضــیکې منطقي برخه (ALU(Arithmetic logic unit)) دیزاین کیږي.

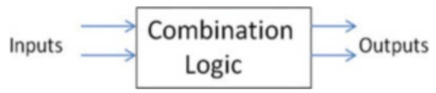
BCD

استعمال کړئ د اوو خطونو اصلي بڼې ته بدلوونکی (Seven-Segment Decoder) له پاره چې عدد د لسو په قاعدې (Decimal) ښکاره کړي.

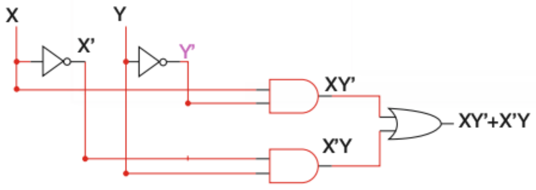
۱.۴ پیژندنه

ډیجېټل سرکټ په ترکیبي (Combinational) او مسلسل (Sequential) منطق باندې طبقه بندی شوی دی. ترکیبي (Combinational) منطق یو داسې ډیجیټل سرکټ دی چې یو یا زیات ورکړیز (Input) او راکړیز (Output) معلومات لري. دغه دیجیټل سرکټ یو مشخص کار ترسره کوي، چې راکړیز (Output) یې د ورکړیز په اوسني قیمت پورې اړه لري. او دا یو بې حافظې سرکټ ده، مسلسل (Sequential) منطق د حافظې (Memory) درلودونکی سرکټ دی. ۱.۴ شکل یوبلاک دیاگرام د ترکیبي (Combinational) منطق ښکاره کوي د څو ورکړیز (Inputs) او څو راکړیز (Outputs) معلوماتو سره. په ترکیبي (Combinational) منطق کې راکړیز (Output) د ورکړل شوي ورکړیز (Input) دنده ده.

فرض کړئ چې F(X,Y)= Xy' + X'y کوم چې په ۲.۴ انځور کې راکړل شوي دي، دغه منطقي ګېټ د انډ (AND)، ار (OR) او ناټ (NOT) ګېټونو څخه جوړشوی دی. د ترکیبي (Combination) منطق راکړیز (Output) د هغه د ورکړیز (Input) په تغیر سره تغیر کوي. ۱.۴ جدول د ۲.۴ انځور له پاره صدق جدول (Truth Table) ښکاره کوي.



۱.۴ انځور: د ترکیبي (Combinational Block) منطق بلاک دیاګرام.



۲.۴ انځور: ترکیبی منطقي سرکټ

X	Y	XY'	X'Y	F
0	0	0	0	0
0	1	0	1	1
1	0	1	0	1
1	1	0	0	0

۱.۴ جدول: د ۲.۴ انځور له پاره صدق جدول (Truth Table).

۲.۴ د ترکيبي منطق تحليل (Analysis of Combinational Logic)

د ترکيبي(Combinational) منطق (Logic) د تحليل اصلي دليل د ترکيبي (Combinational) منطق د راکړيز(Output) د دندې او صدق جدول (Truth Table) پيدا کول دي. ۳.۴ انځور يو ترکيبي (Combinational) منطق ښکاره کوي. د ترکيبي (Combinational) منطق راکړيز (Output) د F (Function) په واسطه ښودل کيږي.او ۲.۴ انځورF فنکشن له پاره صدق جدول (Truth Table) ښرگندوي.

T1= [X+Z]'+Xy]'=(Xy)'

(X+Y)(Xy)=(X+Z)(X'+Y')=Xy'+X'z+Y'z

$$F(X, Y, Z) = (X + Z)\,T1 = (X + Z)(XY' + X'Z + Y'Z)$$

$$= XY' + XY'Z + XY'Z + X'Z + Y'Z$$

$$F(X, Y, Z) = XY' + XY'Z + X'Z + Y'Z = XY' + Y'Z(X + 1) + X'Z$$

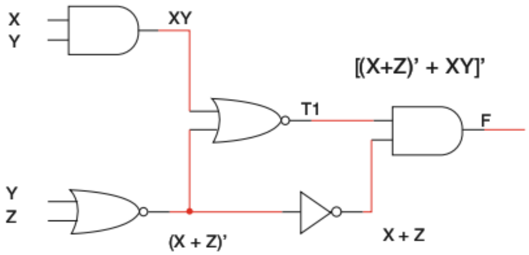
$$= XY' + Y'Z + X'Z$$

۳.۴ د ترکيبي منطق ډيزاين (Design of Combinational Logic)

۴.۴انځور د ترکيبي منطق (Combinational Logic) له پاره بلاک دياگرام ښرگندوي، لاندې مراحل دا ښرگندوي چې څنگه کولای شو يو ترکيبي منطق

(Combinational Logic) ډیزاین کړو:

۱. د هغو مشـــکلاتو په نښـــه کول چې دترکیبي منطق (Combinational Logic) دندی تشریح کوي.



۳.۴ انځور د ترکیبي منطق (Combinational Logic)

X Y Z	XY'	Y'Z	XZ'	F
0 0 0	0	0	0	0
0 0 1	0	1	0	1
0 1 0	0	0	0	0
0 1 1	0	0	0	0
1 0 0	1	0	1	1
1 0 1	1	1	0	1
1 1 0	0	0	1	1
1 1 1	0	0	0	0

۲.۴ جدول: د ۳.۴ انځور صدق جدول (Truth Table)

۴.۴ انځور د ترکیبي منطق (Logic Combinational) بلاک دیاګرام

۲. د راکړیزو (Inputs) او ورکړیزو (Outputs) تعداد مشـخص کول، یا شـاید دا د مشکلاتو د څرګندولو وخت کې مشخص شي.

۳. ورکړیز (Input) او راکړیز (Output) ته د متحول په نښه کول.

۴. د ټولو ورکړیزو (Inputs) پیـدا کول او د هغو څخـه د صـــدق جدول (Truth Table) جوړول ، او په نتیجه کې راکړیز (Outputs) د مشکلاتو په څېړلو سره تر لاسه کېږي.
۵. د کارناف نقشې (K-Map) څخه په استفادی د راکړیز (Output) تابع لیکل.
۶. د منطقي (Logic) سرکټ رسمول.

۱.۴ مثال: یو ترکیبي منطق (Combinational Logic) سـرکټ دیزاین کړئ چې درې ورکړیز (Inputs) او یو راکړیز (Output) ولري، کله چې د ورکړیز (Input) شــمیر د درېو څخه زیات او یا مســاوي شــي نو راکړیز (Output) یو 1 کیږي ،په معکوس حالت کې راکړیز (Output) د صفر (0) سره مساوي کیږي.

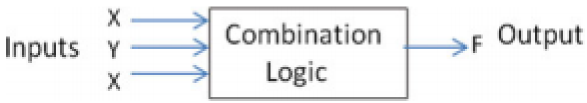
حـل : ۴.۴ انځور یو ترکیبي منطق (Combinational Logic) څرګنـدوي چې درې ورکړیز (Inputs) او یو راکړیز (Output) لري.

د Y، X او Z متحولین ورکړیز (Inputs) په نښــه کوي، او F د راکړیز (Output) تابع په نښــه کوي. د ۳.۴ انځور د ۵.۴ انځور له پاره یو صـــدق جدول (Truth Table) څرګندوي.

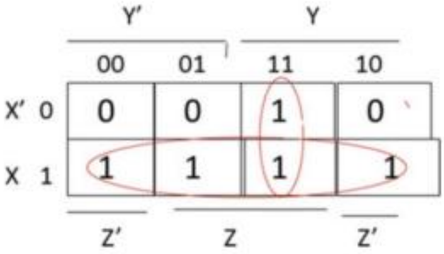
په ۳.۴ جدول کې د ټول Y، X اوz ورکړیز (Inputs) ترکیبونه لیســت شــوي دي. او د مشــکل د راپور له مخی راکړیز (Output) (F) مســاوي کیږي د یو (1) ســره کله چې د ورکړیزو (Inputs) شمیر درې یا د هغه څخه زیات وي.

XYZ	F
000	0
001	0
010	0
011	1
100	1
101	1
110	1
111	1

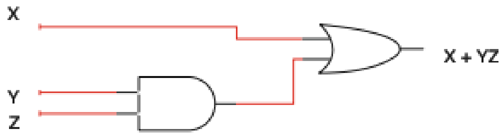
۳.۴ جدول: د لومړی مثال صدق جدول(Truth Table)



۵.۴ انځور د څلورم مثال د ترکيبي منطق (Combinational Logic) بلاک دیاگرام.



٦.۴ انځور: د څلورم مثال کارناف نقشه (K-Map)

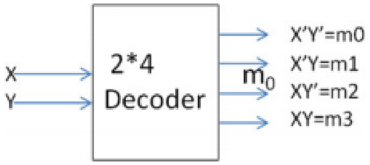


۷.۴ انځور: د څلورم مثال له پاره سرکټ دیاګرام

که نه نو د هغې راکړيز (Output) مساوي کيږي د صفر (0) سره. د راکړيز (Output) تابع F د مينټرم (Minterm) د مجموعې په ډول هم ښودل کيږي.

F(X,Y,Z) = m3 + m4 + m5 + m6 + m7 (Sum Of Minterms)

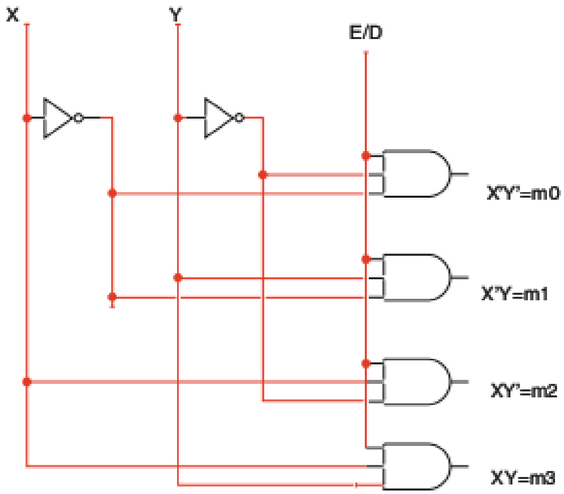
د F(X,Y,Z) Minterms اړول کارناف نقشې (K-Map) ته لکه څرنګه چې په ٦.۴ انځور کې ښودل شوي او د کارناف نقشې (k-map) څخه د F(X,Y,Z) = X + YZ ساده تابع لاسته لاوړل شوې، ۷.۴ انځور يو منطق (Logic) سرکټ ښکاره کوي د F تابع له پاره (۷.۴ انځور).



۸.۴ انځور: د بلاک دیاگرام ډیکوډر ۴x ۲ (Decoder)

X Y	m_0	m_1	m_2	m_3
0 0	1	0	0	0
0 1	0	1	0	0
1 0	0	0	1	0
1 1	0	0	0	1

۴.۴ جدول: د ۲x۴ ډیکوډر (Decoder)



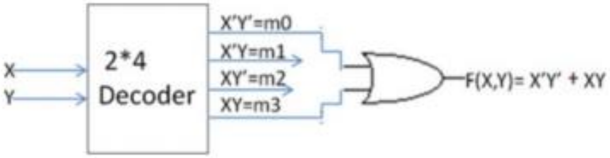
۹.۴ انځور: منطقي سرکټ ۲x۴ډیکوډر(Decoder) د E/D سره

۴.۴ اصلي بڼې ته بدلوونکی (Decoder)

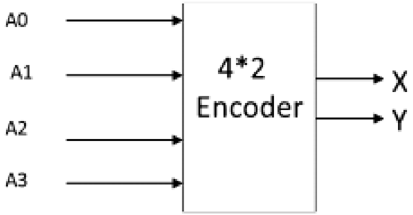
اصـلي بڼې ته بدلوونکی (Decoder) یو MSI (د ډېرو ترکیبي منطق څخه جوړ شوی سرکټ) منطق دی چې یو ټولګي ورکړېز له پاره مین ټرم (Minterm) جوړوي، د X او Y دوه متحولین(څلور مینټرمونـه Minterms) جوړوي، چې عبـارت دي لـه $X'Y' = m_0$، $X'Y = m_1$، $XY' = m_2$ او $XY = m_3$ ۸.۴ انځور د۲x۴ اصلي بڼې ته بدلوونکی بلاک دیاګرام ښکاره کوي(دوه ورکړېز او څلور راکړېز). ۴.۴ جدول د۲x۴ اصلي بڼې ته بدلوونکې صدق جدول (Truth Table) ښکاره کوي، لاندې تابع اصلي بڼې ته د بدلوونکې یو راکړېز دی :

$$M_0 = X'Y', m_1 = X'Y, m_2 = XY' \text{ and } m_3 = XY$$

د ۹.۴ انځور د ۲*۴ اصـــلي بڼې تـه بـدلوونکي(Decoder) منطقي(Logic) سـرکټ ښـکاره کوي. زیاتره MSI ICs یو اضـافي ورکړېز(Input) لري چې د فعال او غیرې Enable/Disable (E/D) په نوم یادېږي ورکړېز لرونکې تابع یو.



۱۰.۴ انځور: د اصلي بڼې ته بدلوونکي(Decoder) په وسیله یو تابع نطبیق کوي



۱۱.۴ انځور: د۲x۴ بل حالت ته بدلوونکي (Encoder) بلاک دیاګرام

په ۹.۴ انځور کې ښودل شوي، کله چې E/D = 0 نو د اصلي بڼې ته بدلوونکي ټول IC حاصل د صفر سره مساوي کيږي. اصلي بڼې ته بدلوونکې يي غير فعال(Disable) دی، اصـــلي بڼې تـه بدلوونکی په هغه صـــورت کې مين ټرم (Minterm) توليدوي کله چې E/D = 1 وي.

۱.۴.۴ د اصـــلي بڼې تـه د بـدلوونکې څخـه پـه اســـتفـادې د تـابع تطبيق کول (Implementing a Function Using a Decoder): اصـلي بڼې ته

بدلوونونکي Decoder د ترکيبي (Combinational) سرکټ د ډيزاين له پاره هم اســـتفاده کيږي، فرض کړئ چې $F(X,Y) = XY + X'Y'$ يو تابع ده چې د اصـــلي بڼې ته د بدلوونکې (Decoder) څخه په اســتفادی تطبيق کېدای شـــي، تابع د $F(X,Y) = m_3 + m_0$ په شـــکل هم ښـــودل کيږي، تابع دوه متحولين لري چې د اصلي بڼې ته د بدلوونکي له پاره دوه ورکړيز (Inputs) دي، نو ددی له پاره 4*2 اصـــلي بڼې ته بدلوونکي (Decoder) ته اړتيا ده او په لاس راغلې تابع يي د m_3او m_0 مجموعه ده، لکه چې په ۱۰.۴ انځور کې ښودل شوې ده.

۵.۴ بل حالت ته بدلوونکې (Encoder)

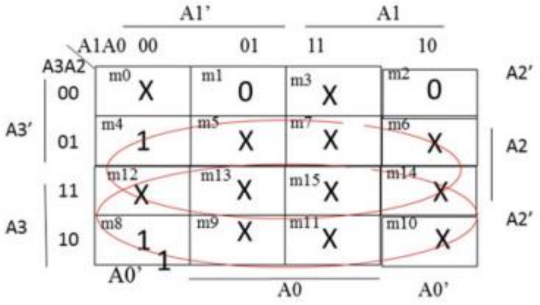
بل حالت ته بدلوونکی (Encoder) د اصلي بڼې ته د بدلوونکی (Decoder) معکوس دی، دا 2^n ورکړيز او n راکړيز لري، يعنی د n=2 په صورت کې بل حالت ته بدلوونکې (Encoder) د 2^n فورمول څخه په ګټې اخيستنې سره 4 ورکړيز او 2 راکړيز لري. ورکړل شوي ورکړيز يو د دوه په قاعدې قيمت په لاس ورکوي. ۱۱.۴ انځور د۲x۴بل حالت ته بدلوونکي Encoder بلاک ډياګرام ښکاره کوي. ۵.۴ انځور د۲x۴ بل حالت ته بدلوونکی (Encoder) صدق جدول (Truth Table) ښکاره کوي. په ۵.۴ انځور کې که چيري A3=1 شـی نو حاصل يي Xy = 11 يا که چيري A2 = 1 شی نو Xy = 10 سره کيږي.

۱۲.۴ انځور د X او Y تابع له پاره کارناف نقشـه (K-Map) ښکاره کوي، هغه ورکړيز

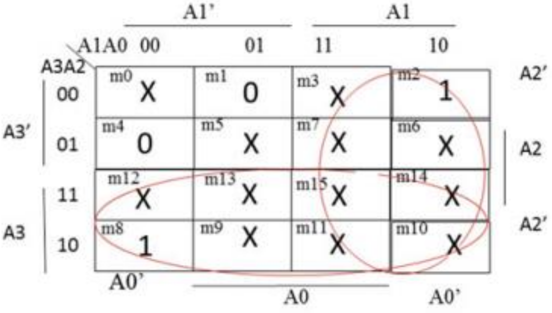
(Inputs) ترکیبونـه چې پـه۲x۴ اصـلي بنې تـه بـدلوونکې په صـدق جدول (Truth Table) کې ندی لیست شـوی د اندیښنـې وړ نده ، په دواړه کارناف نقشـو(K-Map) کې د 'X' سـره ښـودل شـوی، ۱۳.۴ انځور۲x۴ اصـلي بنې ته بدلوونکې له پاره منطقي سرکټ ښکاره کوي.

A3	A2	A1	A0	X	Y
0	0	0	1	0	0
0	0	1	0	0	1
0	1	0	0	1	0
1	0	0	0	1	1

۵.۴ جدول: د۲x۴ بل حالت ته بدلوونکی (Encoder) صدق جدول



X= A3 +A2

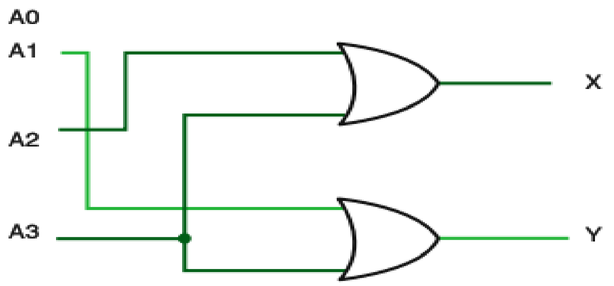


Y = A1 + A3

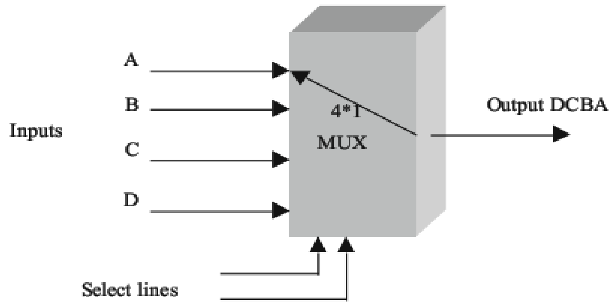
۱۲.۴ انځور: د X او Y تابع د کارناف نقشه K-Map

۶.۴ ملټيپليکسر (Multiplexer) (MUX)

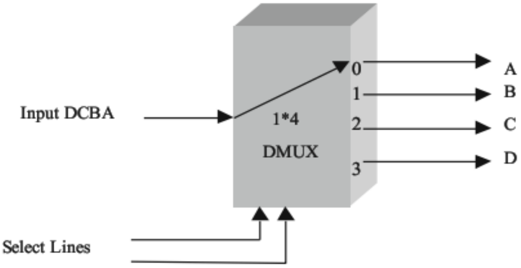
MUX یو ترکیبي (Combinational) منطقي سـرکټ دی چې N ورکړیز (Inputs) او یو راکړیز (Output) لري، د MUX وظیفـه داده چې د څو ورکړیزو (Inputs) څخه یو انتخابوي او راکړیز (Output) ته یي مسـتقیمه لاربـونه کوي. د۱۴.۴ انځور د ملټيپليکسـر (Multiplexer) یو سـاده جوړښت ښکاره کوي. یو ملټيپليکسر د N ورکـړیز (Inputs) او یو راکړیز (Output) سـره د n -څخـه تر- ۱ ملټيپليکسـر (Multiplexer)په نوم یادیږي.



۱۳.۴ انځور: د ۲*۴ بل حالت ته بدلوونکي (Encoder) منطقي سرکټ



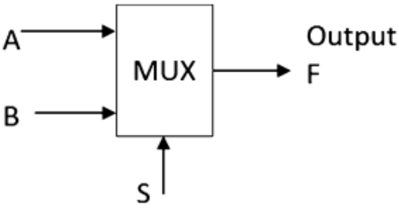
۱۴.۴ انځور: ۴-څخه تر-۱ ملټيپليکسر (Multiplexer)



۱۵.۴ انځور: د ۱-څخه تر-۴ ډی ملټيپليکسر (Demultiplexer)

په يو وخت کې يو ورکړيز انتخابوي او راکړيز ته يي انتقالوي، کله چې سـويچ په A موقعيت کې وي، نو دا A ورکړيز راکړيز ته انتقالوي، او کله چې B موقعيت ته منتقل شي نو هغه B ورکړيز راکړيز ته انتقالوي. او دا ميتود تر هغه ادامه پيدا کوي چې سويچ د D موقعيت ته منتقل شي او D ورکړيز راکړيز ته انتقال کړي.

د ملټيپليکسر (Multiplexer) مخالف د ډيملټيپليکسر (Demultiplexer)(DMUX) دی، چې په ۱۵.۴ انځور کې ښودل شوی دی. په دې کې هر ورکړيز ته د هغی مربوطه راکړيز ته انتقالیږي .



۱٦.۴ انځور: د ۲x۱ MX بلاک دياګرام

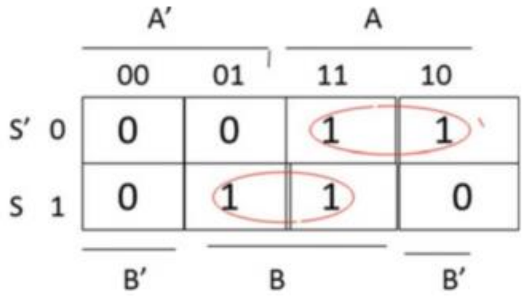
S	A	B	F
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

۶.۴ جدول: د ۲x۱ MUX صدق جدول

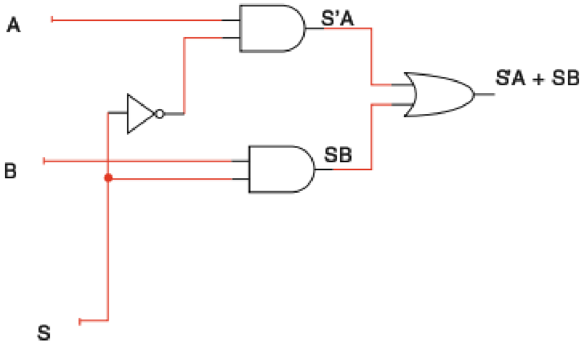
ډیملټیپلکســـر یو ورکړیز او N راکړیز معلومات لري – چې دا د 1 په N کې ډیملټیپلیکسـر (Demultiplexer) په نوم یادیږي. کله چې سـویچ په (0) موقعیت کې وی نو دا A انتقالوي د (0) ورکړیز موقعیت (Port) ته او بیا حرکت کوي د (۱) ورکړیز موقعیـت (Port) تـه او B انتقـالوي هغه موقعیت ته. او دا جریان پیدا کوي تر هغه چې سویچ د دریم ورکړیز موقعیت ته انتقال شی او D دریم موقعیت ته انتقالوي.

۱۶.۴ انځور د ۲x۱ MUX شــکاره کوي چېرتـه چې A او B دوه راکړیز او S یو ټاکل شوی لاین دی، کله چې S = 0 وی نو د ملټیپلکسر ورکړیز د A قیمت ده، کله چې S = 1 شي نو د ملټیپلکسر ورکړیز د B قیمت دی.

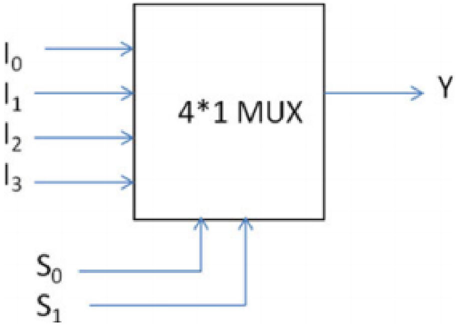
$$F(S, A, B) = m_3 + m_5 + m_7$$



۶.۴ جدول: و ۱۶.۴ انځور MUX صدق جدول (Truth Table) څرګندوي.



۱۷.۴ انځور: د ۲x۱ MUX لاجیک سرکټ



۱۸.۴ انځور: د ۴x۱ MUX بلاک دیاګرام

S_0	S_1	Y
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

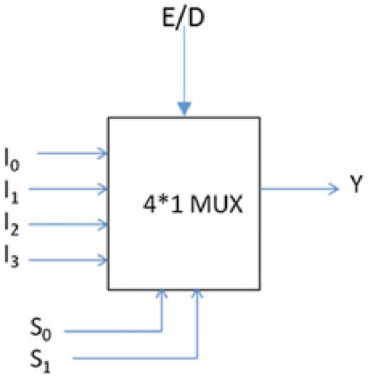
۷.۴ جدول: د MUX عملیه

د کارناف نقشې (K-Map) د لوستلو په نتيجه کې $F(S, A, B) = S'A +$ SB . ۱۷.۴ انځور د ۲x۱ MUX لاجيک سرکټ ښکاره کوي.

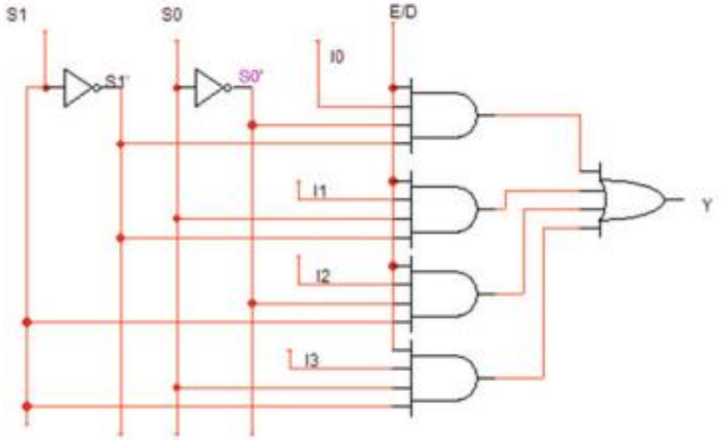
۱۸.۴ انځور د ۴x۱ MUX ښکاره کوي چې I_2, I_1, I_0 او I_3 راکړيز دي ، Y ورکړيز دی ، او S_0 او S_1 ټاکل شوی لاينونه دي. د ۱۱.۴ جدول د MUX عمليه ښکاره کوي. د Y تابع د ۷.۴ جدول څخه لاسته راوړلی شو ، کله چې ورکړيز $S_0S_1 = 00$ وي نو $Y = I_0$ ، کله چې $S_0S_1 = 01$ نو $Y = I_1$ او،کله چې $S_0S_1 = 10$ نو $Y = I_2$ او کله چې $S_0S_1 = 11$ نو $Y = I_3$ ، نو په دې صورت کې د Yراکړيز عبارت دي له :

$$Y = S_0' S_1' I_0 + S_0' S_1 I_1 + S_0 S_1' I_2 + S_0 S_1 I_0$$

۱۹.۴ انځور د ۴x۱ MUX سره د E/D بلاک دياګرام ښکاره کوي. ۲۰.۴ انځور د ۴x۱ MUX منطقي سرکټ ښکاره کوي. لکه چې په شکل کې ښودل شوي، E/D راکړيز د MUX لاجيک دياګرام اضافه کېږي، کله چې $E/D = 0$ نو ورکړيز Y=0 سره او MUX کمزوری غير فعال (Disable) کېږي.



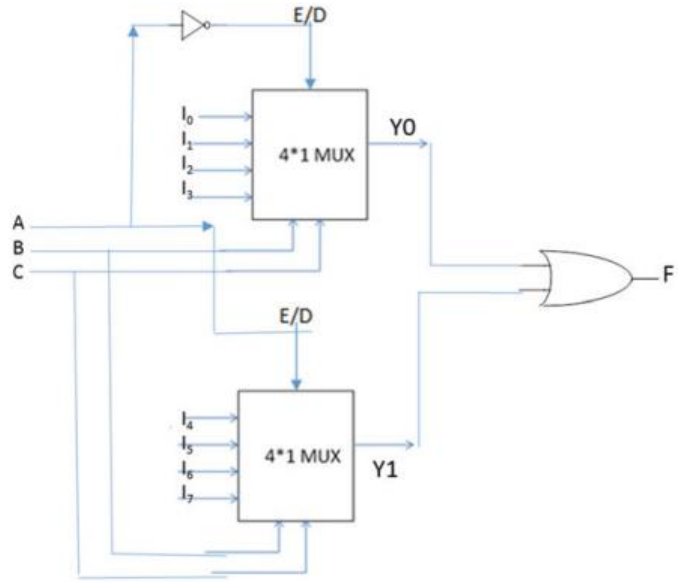
۱۹.۴ انځور: د ۴x۱ MUX سره د E/D بلاک دياګرام ښکاره کوي.



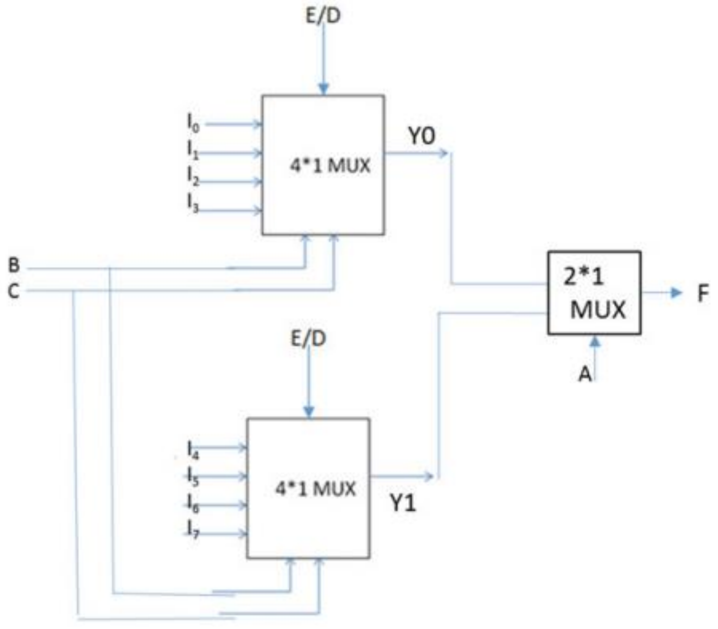
۲۰.۴ انځور: د ۴x۱ MUX لاجیک سرکټ دیاګرام

۱.۶.۴ د کوچنی ملټیپلکسر څخه په استفادی د لوی ملټیپلکسر ډیزاین کول (Designing Large Multiplexer Using Smaller Multiplexers):

د کوچنیو MUXs په یو ځای کولو سره کولی شو یو لوي MUX لاسته راوړو، یو ۸x۱ MUX د دوه ۴x۱ MUX او یو ار (OR) ګیټ د یو ځای کېدو څخه لاسته راوړلی شو، لکه چې په ۲۱.۴ انځور کې ښودل شوې دی. په دغه شکل کې A,B او C ټاکل شوي لاینونه دي. کله چې A = 0 شي نو پورتنی MUX فعال کېږي، او کله چې A = 1 شي نو لاندنی MUX فعال کېږي. همدارنګه د ۸x۱ MUX د دوه ۴x۱ MUX او یو ۲x۱ MUX څخه هم تر لاسه کېدای شي لکه چې په ۲۲.۴ انځور کې چې ښودل شوي. په دې ۲۲.۴ انځور کې ، کله چې A = 0 شی نو راکړیز معلومات یې F = Y0 سره او کله چې A = 1، شی نو راکړیز معلومات یې F = Y1 سره.



۲۱.۴ انځور: د دوه ۴x۱ MUX او یو اور(OR) ګیټ څخه په استفادې د ۸x۱ MUX ډیزاین



۲۲.۴ انځور: یو ۸x۱ MUX د دوه ۴x۱ MUX او یو ۲x۱ څخه لاسته راغلي.

۲.٦.۴ د ملټیپلکسر څخه په استفادې د تابع اجرا کول (Implementing Functions Using Multiplexer):

a) د ۸x۱ MUX څخه په استفادې د درې متحوله صدق جدول (Truth Table) جوړول:

Mux د ډیجیټل تابع د تر لاسه کولو له پاره هم استفاده کېدلی شي. ۸.۴ جدول کی صدق جدول (Truth Table) فرض کړئ، دغه صدق جدول (Truth Table) د درې متحولونو څخه جوړشوې دی ، او دا صدق جدول (Truth Table) د ۸x۱ MUX څخه په استفادی اجرا کېدای شي (۸.۴ جدول).

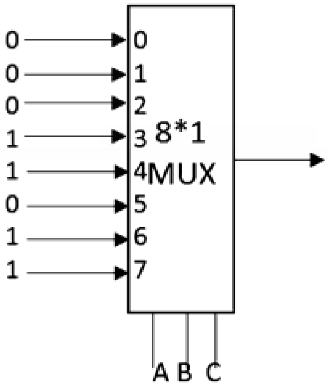
دغه صدق جدول (Truth Table) د درې متحولونو څخه جوړشوی، نو په دې صورت کې یو ۸x۱ MUX ته ضرورت لري، د B, A او C متحولونو د MUX د ټاکلو لاینونو (Select Lines) سره وصل شوي ، او د MUX ورکړیز صدق جدول د (Truth Table) د راکړیز سره مطابقت ورکول کېږي (۲۳.۴ انځور).

b) د ۸x۱ MUX په استفاده د ۸.۴ جدول جوړول:

د متحولونو تعداد -۱ = د انتحاب شـویو لاینونو تعداددرې متحولین ۱ = ۲ د MUX له پاره د انتخاب شـویو لاینونو تعدادکه چیري A او B د انتخاب شـوي لاینونو سره وصل وي، کله چې AB = 00 وي، نو په ۹.۴ جدول کې دوه سطرونه (Rows) وي د AB = 00 سره ، نو ورکړیز F د C قیمت پورې مربوط وي. په دی صورت C = 0 سره. فرض کړئ چې سطرونه د AB = 01 ، F = 0 د C = 0 له پاره ، او F = 1 د C = 1 له پاره، په دی صورت F = C سره کېږي. ۲۴.۴ انځور د ۴x۱ MUX په واسطه د ۹.۴ م انځور اجرا کول ښایي.

A	B	C	F
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

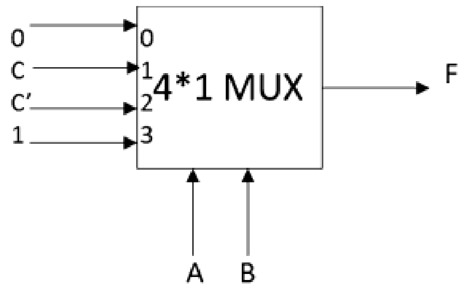
۸.۴ جدول: د درې متحوله MUX له پاره



۲۳.۴ انځور: د ۸x۱ MUX بلاک دیاګرام

A	B	C	F	
0	0	0	0	0
0	0	1	0	
0	1	0	0	C
0	1	1	1	
1	0	0	1	C′
1	0	1	0	
1	1	0	1	1
1	1	1	1	

۹.۴ جدول د ۲۴.۴ انځور له پاره



۲۴.۴ انځور: د جدول ملټيپلکسر (Multiplexer)

۷.۴ نیم جمع کوونکی ، پوره جمع کوونکی ،دوه یز جمع کوونکی او تفریق کوونکی
Half Adder, Full Adder, Binary Adder, and) (Subtractor

نیمه جمع کوونکی (HA) دا یو منطقي سرکټ دی چې د X او Y بیټونه (Bits) سره جمع کوي. ۲۵.۴ انځور د HA بلاک دیاګرام ښکاره کوي، د HA له پاره راکړیز د X او Y دي، او د HA راکړیز د S مجموعه او C حاصل (Carry) څخه عبارت ده. د ۱۰.۴ جدول د HA له پاره صدق جدول (Truth Table) ښکاره کوي. په دی صدق جدول (Truth Table) کې کله چې X = Y = 1 شي نو X جمع Y = 10نتیجه S = 0 او حاصل1 = c کېږي.

د S او C تابعګانې:

S = m1 + m2 = X’Y + XY’ = X XOR Y

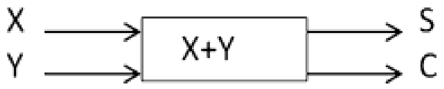
C = XY = X AND Y

۲٦،۴ انځور: د HA له پاره منطقي سرکټ ښکاره کوي.

پوره جمع کوونکی(Full Adder): ۲۷.۴ انځور د FA لـه پـاره بلاک ديـاګرام ښــکـاره کوي. FA د X + Y+ Cin (هر يو يي يواځې (۱) بيټ Bit کېدای شــی) ، د FA راکړيز د S او Cout په نوم ياديږی، او ۱۱.۴ جدول د FA له پاره جدول ښــکاره کوي. په دی جدول کې X، Y او Cin سـره جمع کيږي او په نتيجه کې مجموعه S او حاصـل (Cout) لاســته راځي د S تابع د مينترمونو(Minterms) د مجموعې په شــکل ســره ښودلاې شو لکه څرنګه چې په ۱.۴ معادله کې ښودل شوی دی.

S(X, Y, Cin) = X'Y'Cin + X'YCin' +X'YCin + XY'Cin'+ XYCin

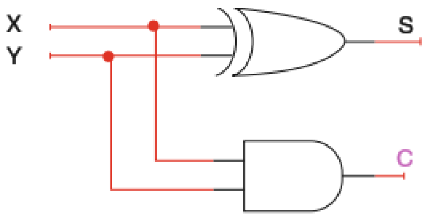
يا



۲۵.۴ انځور: د HA بلاک دياګرام

X Y	C	S
0 0	0	0
0 1	0	1
1 0	0	1
1 1	1	0

۱۰.۴ جدول: د HA لپاره د صدق جدول



۲٦.۴ انځور: د HA له پاره منطقي سرکټ

۲۷.۴ انځور: د HA بلاک دیاګرام

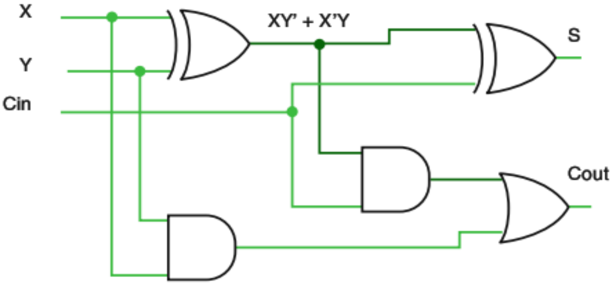
X Y Cin	Cout	s
0 0 0	0	0
0 0 1	0	1
0 1 0	0	1
0 1 1	1	0
1 0 0	0	1
1 0 1	1	0
1 1 0	1	0
1 1 1	1	1

۱۱.۴ جدول: د HA له پاره د صدق جدول

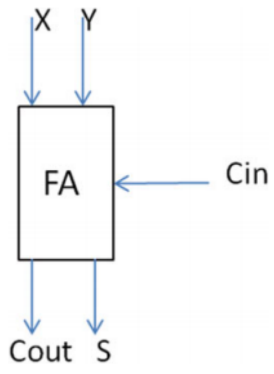
$$(4.2)\ S(X,Y,Cin) = Cin\ (X'Y'\ +\ XY) +\ Cin'(X'Y\ +\ X'Y)$$
$$X'Y\ +\ X'\ =\ X\ XOR\ Y\ =\ A$$
$$X'Y\ +\ XY\ =\ (X'Y\ +\ X'Y)'\ =\ A$$

نو په دی صورت د ۲.۴ معادله د ۳.۴ معادلې په شکل هم لیکلی شو:

$$(4.3)\ \ S(X,Y,Cin)\ =\ CinA'\ +\ Cin'A\ =\ Cin\ XOR\ A$$



۲۸.۴ انځور: د FA منطقي سرکټ



۲۹.۴ انځور: د FA بلاک دیاګرام

(Cout) د مینټرمونو (Minterms) د مجموعې په ډول هم لیکل کېدای شي:

$$Cout(X, Y, Cin) = X'YCin + XY'Cin + XY'Cin + XYCin' + XYCin$$

یا:

$$Cout(X, Y, Cin) = Cin(X'Y, + XY') + XY(Cin' + Cin)$$

یا:

$$Cout(X, Y, Cin) = Cin(X'Y, + XY') + XY$$

۲۸.۴ انځور د پوره جمع کوونکي (Full Adder) له پاره منطقي دیاګرام خرګندوي.

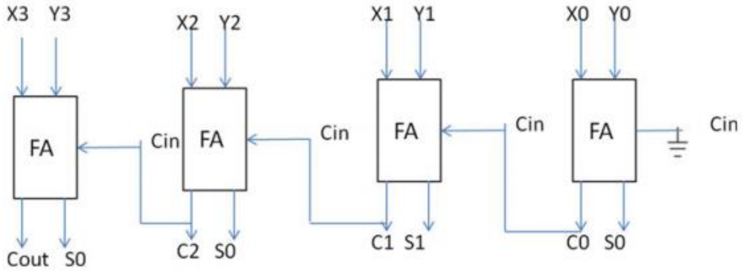
۲۹.۴ انځور د FA له پاره بلاک دیاګرام خرګندوي، کله چې X, Y او Cin ورکړیز دی او S او Cout راکړیز دي.

۱.۷.۴ څلور بیټه دوه یزه جمع کوونکی (4-Bit Binary Adder): د ۴-بیټ دوه یزه جمع کوونکي وظیفه داده چې ۴-بیټ نمبرونه سره جمع کوي لکه:

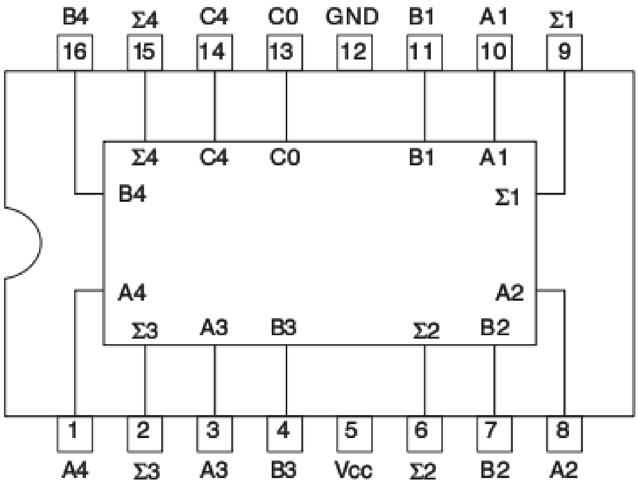
$X_3X_2X_1X_0 + Y_3Y_2Y_1Y_0$

کله X_0 او Y_0 سره جمع کوي په نتیجه کې مجموعه S_0 او حاصل C_0 لاسته

راځي، C_0 بیا د X_1 او Y_1 سره جمع کېږي او په نتیجه کې S_1 او C_1 او لاسته راځي.
۳۰.۴ انځور یو ۴-بیټ دوه یزه جمع کوونکی ښکاره کوي. Cin د ګراونډ سره وصل کېږي چې صفر (0) ښکاره کړي.
د ۴-بیټ دوه یزه جمع کوونکی جوړښوی دیاګرام په ۳۱.۴ انځور کې د یو مدغم شوی سرکټ (IC) په شکل ښودل شوی، او د مدغم شوی سرکټ (IC) نمبر ۷۴۸۳ ده.



۳۰.۴ انځور: ۴-بیټ دوه یزه جمع کوونکی



۳۱.۴ انځور: ۷۴۸۳ م ۴-بیټ جمع کوونکی

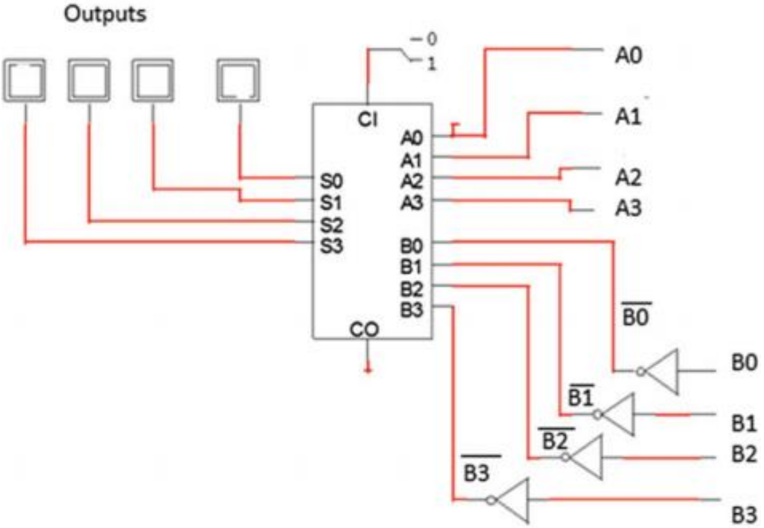
۲.۷.۴ تفریق کوونکی (Subtractor): د تفريق کوونکی د A-B يا A + B + 1 د تفريق عمليه اجرا کوي. ۲۳.۴ انځور د يو منفي کوونکی دياګرام د ۴-بيټ دوه يز جمع کوونکی (Binary Adder) څخه په استفاده ښايي. CI د 1 ته ايښودل کيږي، او B0, B1, B2 او B3 برعکس (Complement) کيږي. ۳۳.۴ انځور د ۳۲.۴ انځور بدلون موندلی دی چې دواړه جمع او تفريق عمليي ترسره کوي. CI د ۴-بيټ دوه يز جمع کوونکی (Binary Adder) د کړی اين (Carry In) او CO يي د کړی اوټ (Carry Out) څخه عبارت ده.کله چې د جمع يا منفي سويچ صفر ته کېښودل شي نو دا د جمعې، او که د جمع يا منفي سويچ يو ته کېښودل شي دا د تفريق عمليه ترسره کوي.

۸.۴ حسابي منطقي برخه (Arithmetic Logic Unit)

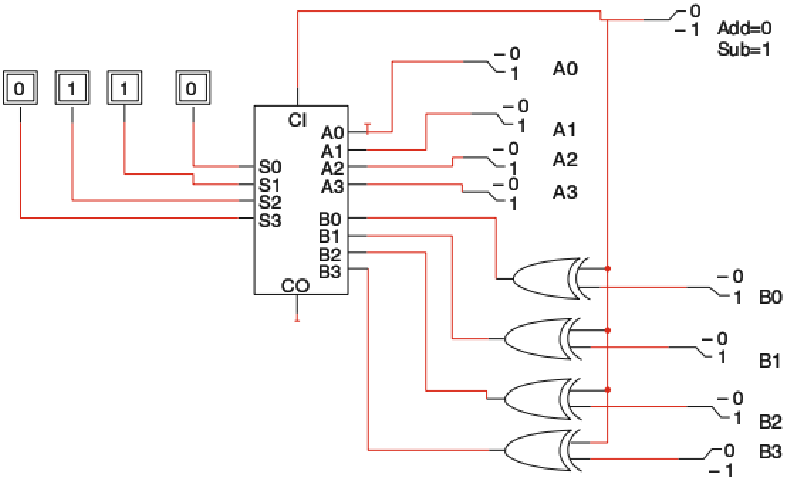
د ALU وظيفه داده چې رياضيکې عمليي لکه د جمعی تفريق او د Bit-Wise منطقي عمليي لکه د انډ (AND)، ار(OR) او ناټ (NOT) عمليي اجرا کوي. ۳۴.۴ انځور د حسابي منطقي برخي بلاک دياګرام ښکاره کوي.

په ۳۴.۴ کې A او B مواصلات (Buses) د حسابي منطقي برخي ورکړيز او C وصلونکی يي راکړيز ده. S1 او S0 يي انتخابونکې لاينونه (Select Lines) دي چې د حسابي منطقي برخي تابعګانې انتخابوي. ۱۲.۴ د حسابي منطقي برخي تابعګانې ښرګندوي. فرض کړئ چې A او B ۴ بيټونه (Bits) دی چې د A3, A2, A1, او A0 او B3, B2, B1 او B0 په شکل ښودل کيږي.

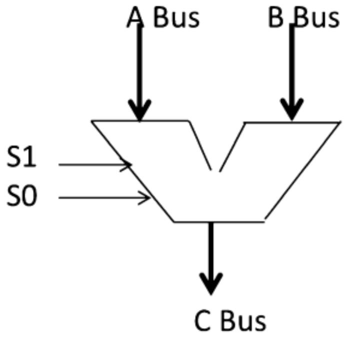
د حسابي منطقي برخي د انتخاب لاينونه د ملټيپليکسر(Multiplexer) اندازه ښرګندوي، څرنګه چې دلته دوه انتخابونکي لاينونه دي، نو ملټيپليکسر(Multiplexer) اندازه ۴x۱ يعنی ۴ ورکړيز او ۱ راکړيز لري، د بيټونو شمير د ملټيپليکسر(Multiplexer) شمير ښکاره کوي. A او B ۴ بيټه دي، نو په دی صورت ۴x۱ ملټيپليکسرونو (Multiplexer) ته ضرورت دی. ۳۵.۴ انځور د حسابي منطقي برخي دياګرام ښرګندوي.



۳۲.۴ انځور: ۴ بیټ تفریق کوونکی



۳۳.۴ انځور: د ۴-بیټ جمع کوونکی(Adder) او منفي کوونکی منطقي سرکټ



۳۴.۴ انځور: د حسابي منطقي برخي بلاک دیاګرام

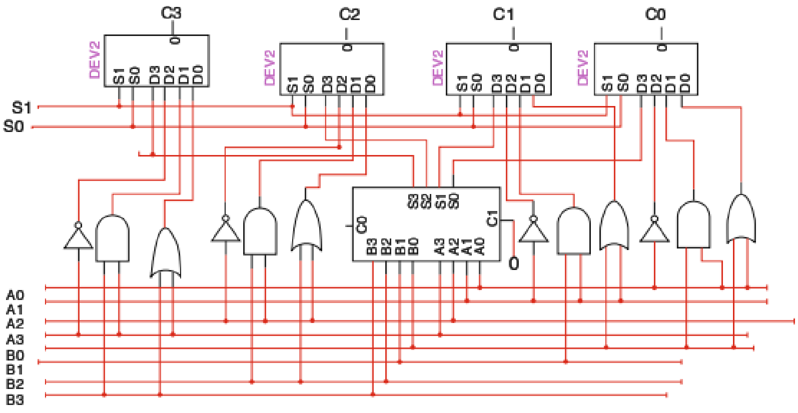
۴,۹ اوو خطو پـه واسـطـه د اعـداو ښـودنـه (Seven-Segment Display)

د اوو خطونو په واسـطه د اعدادو ښـودنه د اووه LED(Light-Emitting Diode) څخه جوړشـوی لکه څرنګه چې په ۳٦.۴ انځور کې ښـودل شوی، اوو خطونو په واسـطه د اعدادو ښـودنه د 1 څخه تر 9 پوری تام عدد ښـکاره کوي ۳۷.۴ انځور هغه برخه ښکاره کوي چې باید په ترتیب سره د صفر څخه تر نهه پوری تام عدد ښکاره کړي.

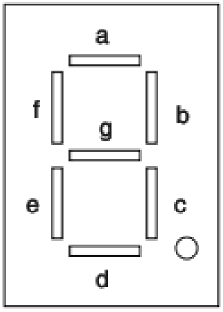
د صفر د ښودلو له پاره ټول برخې بغیر د G څخه باید On وي. د اتو خطونو د ښـــودلو لـه پـاره ټول برخې باید روښـــانه وي. دا یو خاص اصـــلي بڼې ته تبدیلوونکې (Decoder) ته چې د BCD په نوم یادیږی دی ته ضـــرورت لري چې د اوو خطونو په واسطه د اعدادو ښودنه کې دوه یز د لسو په قاعدې کوډ (BCD) د اوو خطونو په واسطه اعدادو ته تبدیل کړي.

S1	S0	ALU
0	0	A OR B
0	1	A + B
1	0	A AND B
1	1	A′

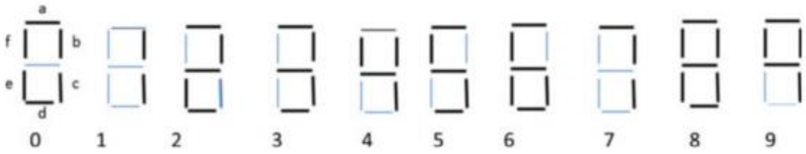
۱۲.۴ جدول: د حسابي منطقي برخه او تابع



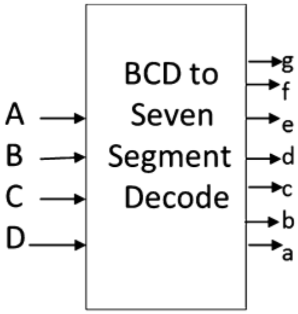
۳۵.۴ انځور: د حسابي منطقي برخي سرکټ ډياګرام



۳٦.۴ انځور: اوو خطونو په واسطه د اعدادو ښودنه



۳۷.۴ انځور: اوو خطونو په واسطه د اعدادو ښودنه کې د (0) څخه تر (9)



۳۸.۴ انځور: د اوو خطونو په واسطه د اعدادو ښودنه کې د BCD بلاک دیاګرام

A	B	C	D	a	b	c	d	e	f	g
0	0	0	0	1	1	1	1	1	1	0
0	0	0	1	0	1	1	0	1	0	0
0	0	1	0	1	1	0	1	1	0	1
0	0	1	1	1	1	1	1	0	0	1
0	1	0	0	0	1	1	0	0	1	1
0	1	0	1	1	0	1	1	0	1	1
0	1	1	0	1	0	1	1	1	1	1
0	1	1	1	1	1	1	0	0	0	0
1	0	0	0	1	1	1	1	1	1	1
1	0	0	1	1	1	1	0	0	1	1
1	0	1	0	d	d	d	d	d	d	d
1	0	1	1	d	d	d	d	d	d	d
1	1	0	0	d	d	d	d	d	d	d
1	1	0	1	d	d	d	d	d	d	d
1	1	1	0	d	d	d	d	d	d	d
1	1	1	1	d	d	d	d	d	d	d

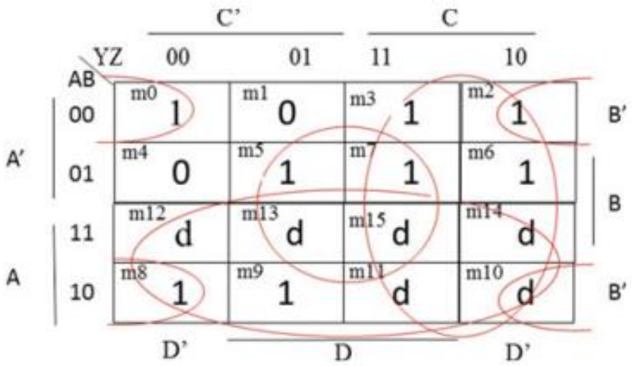
۱۳.۴ جدول: د اوو خطونو په واسطه د اعدادو ښودنه د BCD لپاره جدول)

۳۸.۴ انځور د اوو خطونو په واسطه د اعدادو اصلي بڼې ته تبديلوونکی. (Seven-Segment Decoder) له پاره د BCD بلاک دیاګرام ښکاره کوي. او ۱۳.۴ انځور د اوو خطونو په واسطه د اعدادو اصلي بڼې ته تبديلوونکي (Seven-Segment Decoder) له پاره د BCD جدول څرګندوي. اصلي بڼې ته تبدلوونکې ۴ ورکړېز او ۷ راکړېز لري، د اصلي بڼې ته تبديلوونکی ورکړېز BCD دی کوم چې د

0000 څخه تر 1001 پورې وي، لکه څرنګه چې په ۱۳.۴ جدول کې ښودل شوې، که چیري ورکړیز قیمت د 1001 څخه کم وی، نو راکړیز په نظر کې نه نیول کیږي.
دلته اووه راکړیز دی، چې هر راکړیز یو کارناف نقشې (K-Map) ته ضرورت لري چې د راکړیز تابع پیدا کړی. ۳۹.۴ انځور د A راکړیز لـه پاره د کارناف نقشـه (K-Map) ښکاره کوي.
د ۳۹.۴ م انځور د کارناف نقشې (K-Map) په لوستلو سره نتیجه لاسته راړو چې a تابع :

$$A = B'D' + A + C + BC$$

د پورتنې کړنلارې څخه په استفاده، کولای شو چې نور راکړیز تابعګانې هم لاسته راوړو.



۳۹.۴ انځور: د (a) راکړیز له پاره کارناف نقشه

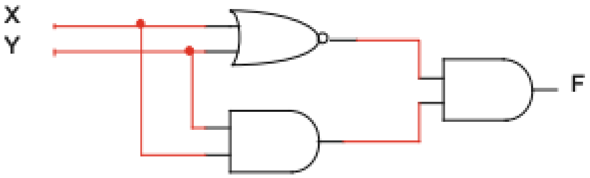
۱۰.۴ خلاصه

- ترکیبي سرکټ یو ډیجیټل سرکټ دی یو یا زیات ورکړیز او یو یا زیات راکړیز لري.
- د ترکیبي سرکټ راکړیز د مربوطه ورکړیز د قیمت پوری اړه لري.
- ترکیبي منطقي سرکټ د تابع یا جدول په واسطه ښودل کېږي.
- اصلي بڼې ته تبدیلوونکې (Decoder) یو ترکیبي منطقي سرکټ دی. چې n ورکړیز او 2^n راکړیز معلومات لري. که چیري $n = 2$ وي نو اصـلي بڼې ته تبدیلوونکی ۲x۴ وي.
- اصــلي بڼې تــه تبـدیلوونکی د (Minterms) ورکړیز تولیـدوي، یو اصــلي بڼې تـه تبدیلوونکی د ۲ ورکړیز سره څلور مینټرمونه (Minterms) جوړوي.
- ملټیپلکسر (Multiplexer) یو ترکیبي سرکټ دی، د 2^n ورکړیز او یو راکړیز سره. چېرته چې n د انتخاب شــوو لاینونو تعداد ښــکاره کوي. که چیري $n = 2$ شــي نو ملټیپلکسر (Multiplexer) باید ۴x۱ وي.
- د نیمه جمع کوونکي (Half Adder) وظیفه داده چې دوه بیټونه ســره جمع کړي مجموعه او حاصل جوړ کړي.
- د پوره جمع کوونکي (Full Adder) وظیفه داده چې درې بیټونه سره جمع کړي.
- د ۴-بیــټ-دوه یز جمع کوونکي (Binary Adder) وظیفـه داده چې دوه ۴-بیــټ نمبرونه سره جمع کړي.
- ریاضـیکې منطقي واحد (Arithmetic Logic Unit) یو ترکیبي سرکټ دی چې ریاضیکې او منطقي عملیې تر سره کوي.
- د اوو خطونو په واسـطه د اعداو ښـودنه د دی له پاره اسـتعمالیږي چې د لسو په قاعده یو عدد وښایي.
- BCD نه تر اوو خطونو پوری اصلي بڼې ته تبدیلوونکي له پاره ۴-بیټ دوه ییز ۷ بیټو ته تبدیلوي خو د اوو خطونو د اعداو د ښودنې سره په تفاوت.
- په پنځم څپرکي کې مسـلسل منطقي سرکټ لوستل کیږي، او هغه موضوعات چې په دی څپرکي کې تـدریس کیږي عبـارت دی لـه: ډی جی کې D,J-K او فلیـپ لاپ (

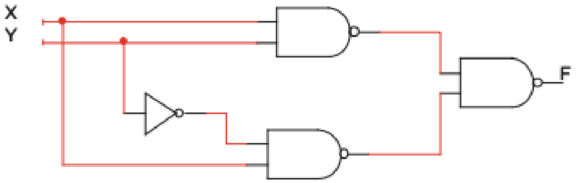
(Flip-Flop)، ريجسټر(Register) ، شيفټ ريجيسټر(Shift Register) ، سټيټ دياګرام (State Diagram) ، سټيټ جدول(State Table) او د حسابوونکی (Counter) ډيزاين کول.

ستونزې

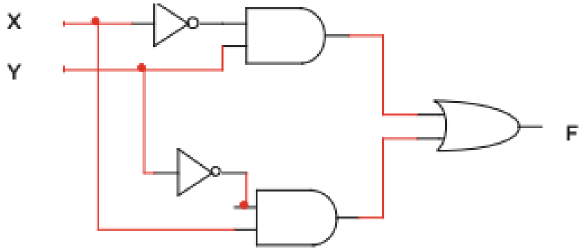
۱. د لاندې ترکیبي سرکټ منطقي ډیاګرام درکړل شوي، تاسو د هر ي تابع راکړیزه تابع او جدول لاسته راوړئ.



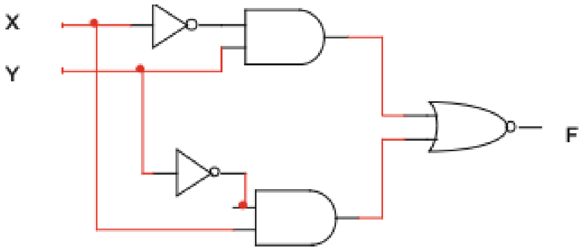
(a



(b



(c

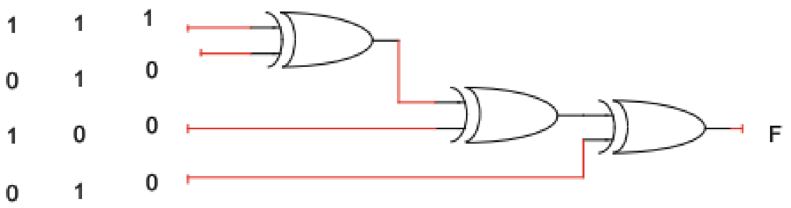


(d

۲. د لاندې ګیټونو راکړیزه تابع لاسته راوړئ.

(a

۳.   د هر ورکړیزله پاره راکړیزه تابع F لاسته راوړئ.



۴.   یو لاجیک سرکټ د درې ورکړیز او یو راکړیز سره ډیزاین کړئ، راکړیز د ورکړیز جفت پیریټي بیټ (Parity Bit) تولیدوي.

a)   جدول یې ښکاره کړئ.

b)   راکړیزه تابع یې پیدا کړئ.

c)   منطقي سرکټ یې رسم کړئ.

۵.   د $F(X, Y, Z) = XY' + XZ'$ تابع اجرا کړئ.

a)   اصلي بنې ته تبدیلوونکی (Decoder)

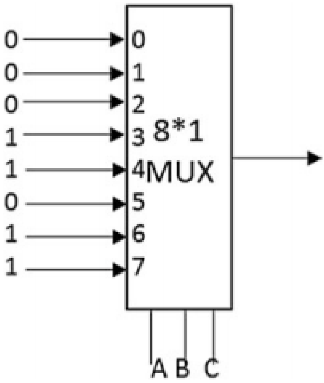
b)   ملتیپلکسر (Multiplexer)

۶.   لاندې تابع یواځې د یو اصلي بنې ته تبدیلوونکي (Decoder) او دیوه خارجی ګیټ په مرسته اجرا کړئ.

$F1(X, Y, Z) = \Sigma(0,3,4)$
$F2(X, Y, Z) = \Sigma(2,3,5)$

۷.   د ډیکوډر په مرسته یو پوره جمع کوونکی (Full Adder) اجرا کړئ.

۸. لاندې ملټيپلکسر (Multiplexer) درکړل شوې دی ، د هغه جدول تکميل کړئ.



A	B	C	F
0	0	0	
0	1	1	
0	1	0	
1	1	1	
1	0	0	
1	0	1	
1	1	0	
0	0	1	

۹. د $F(W, X, Y, Z) = \Sigma(0,1,3,4,7,8,9,11,12,15)$ تابع د MUX په مرسته اجراکړئ.

۱۰. د ۴-بیټ دوه یز جمع کوونکی (4-Bit Binary Adder) څخه په استفادي يو ۸-بیټ دوه یز جمع کوونکی (8-Bit Binary Adder) ډيزاين کړئ.

۱۱. د ۴-بیټ دوه یز جمع کوونکی (4-Bit Binary Adders) څخه په استفادي يو ۱۶-دوه یز جمع کوونکی (16-Bit Binary Adder) ډيزاين کړئ.

۱۲. د درې ورکړيز او درې راکړيز سره يو ترکيبي منطقي سرکټ ډيزاين کړئ ، که چيري ورکړيز 0,1,2 او 3 وي نو 3 راکړيز د ورکړيز څخه زيات وي کـه چيرې ورکړی 4,5,6, يا 7 وي نور 3 راکړيز د ورکړيز څخه کم وي.

۱۳. یو اورګاډی د ۷ غرفو سره مسافر انتقالوي چې هر یو د ۱ څخه تر ۷ پورې لمبر لري، هره غرفه یو دوه ییز سویچ د بیړنی حالت له پاره لري، او هر یو د سویچونو څخه که فعال (On) شـي، نو د غرفی لمبر د کلینډر په خونه کې د لسـو په قاعدې وښـایي. خپل ډیزاین د Logisim پوستغالي په مرسته امتحان کړئ.

۱۴. یو ترکیبي سرکټ د څلور ورکړیز او یو راکړیز سره ډیزاین کړئ، د ترکیبي سرکټ له پـاره ورکړیز په BCD سـره وي،او راکړیز د ورکړیز له پاره حتی جفت پیریټي (Parity) بیټ هم جوړوي.

۱۵. یو ۱x۱٦ MUX د ۱x۴ MUX په مرسته ډیزاین کړئ.

۱٦. یو ۴-بیټ ALU ډیزاین کړئ چې لاندې فعالیت تر سره کړي.

A + B, A – B, A + 1, A’, B’, A OR B, A XOR B, A AND B

۱۷. یو ترکیبي منطقي سـرکټ ډیزاین کړئ چې X او Y سـره مقایسـه کړئ، په داسـی حالکی چې X = X1X0 او Y =Y1Y0 ، د ترکیبي منطق سـرکټ راکړیز یو (۱) وی، که چیري X<Y، وي نو بیا به راکړیز صفر (۰) وي.

a) جدول(Truth Table) ښکاره کړئ.

b) د کارناف نقشې (K-Map) څخه په استفاده د راکړیز تابع پیدا کړئ.

پنځم څپرکی

هم مهاله مسلسل منطق

(Synchronous Sequential Logic)

موخې: ددې فصل په لوستلو سره به تاسې وکولای شئ چې لاندې کارنونه تر سره کړئ:

- د یو ترتیبي منطق تحلیل کول
- په (S-R latch) باندی عملیي ترسره کول
- د (D flip-flop) ډیزاین کول د (S-R latch) څخه
- د (D flip-flop) وظیفې پیژندل او زده کړه
- په (T flip-flop) او (J-K) پیژندل او زده کړه
- د راجستر (Register) ډیزاین کول او راجستر (Register) تـه د (D flip-flop) پواسطه بدلون ورکول
- د ترتیبي سرکټ له پاره یو (State Table) جوړول
- د (State Table) څخه د یو State دیاګرام جوړول
- د هر ډول (Flip-Flop) له پاره د (Excitation Table) جوړول
- رقم شمارونکي یا Digital Counter جوړول

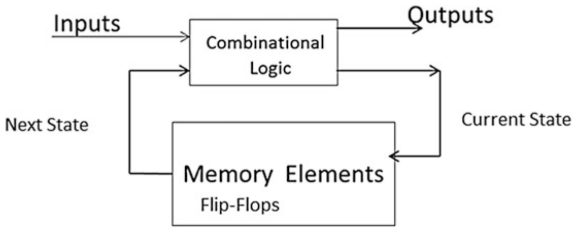
۱.۵ پیژندنه

ترتیبي منطقي سرکټ د حافظې (Memory) عناصر لري، او راکړیز (Output) به يي د ورکړیز (Input) په اوسـنی قیمت اود مخکښی ورکړیز لیول (Input-Level) پوری تړاو لري. ۱.۵ انځور کې د ترتیبي منطق ډیـاگرام ښــودل شــوی چې هر راکړیز (Output) په اوســنی ورکړیز (Input) او د حافظې (Memory) په اوســني حالت پوری تړاو لري. په دغه انځور کې د Combinational منطق هر راکړیز (Output) د حـافظې (Memory) لـه پاره ورکړیز (Input) قیمت دی، او د حافظې (Memory) څخه چې کوم راکړیز(Output) قیمتلاس ته راځي هغه د ترکیبي (Combinational) منطق لـه پـاره ورکړیز (Input) قیمـت دی. د حافظې (Memory) اســاســي اجزاوې (Flip-Flops) دي چې پـه ځـان کې یواځې دوه په قاعده (Binary) قیمتونه ذخیره کولی شي.

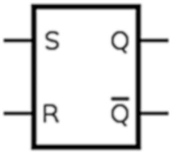
د هم مهال مسـلسل منطق یا (Synchronous Sequential Logic) نتیجه د حافظې (Memory) د عناصــرو په راکړیز (Output) او ورکړیز (Input) واحد پوری تړاو لري.

او د هم مهاله مسلسل منطق (Synchronous Sequential Logic) څخه په لاندې مواردو کې استفاده کیږي:

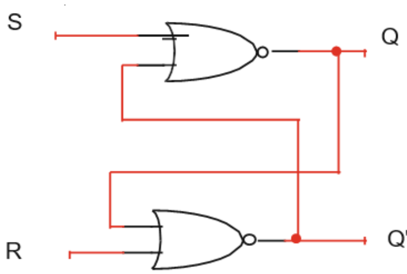
د راجستر (Register) په ډیزاین کولو کې د شمیرونکی (Counter) په جوړولو کې او د حافظې (Memory) په جوړولو کې ور ځینی استفاده کیږي هم مهاله ملسل منطق د ساعت په ضربان یا تیک تک باندی کار کوي.



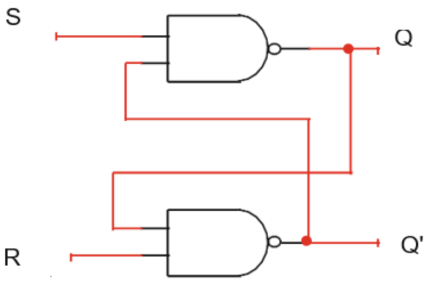
۱.۵ انځور: د همزمان ترتيبي منطق بلاک دیاګرام



۲.۵ انځور: د S-R latch بلاک دیاګرام



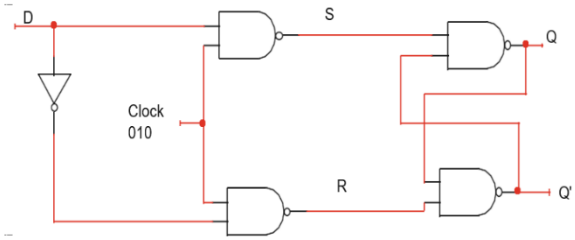
۳.۵ انځور: د S-R latch منطقي سرکټ د NOR ګيټ کارولو سره



۴.۵ انځور: د S-R latch منطقي سرکټ د NAND ګيټ کارولو سره

۲.۵ اس-ار لیچ (S-R Latch)

(S-R Latch) یو ډول حافظه (Memory) ده چې دوه ورکړیز (Inputs) د سیټ S(Set) او ریسیټ R(Reset) سره، او دوه راکړیزو (Outputs) د Q او Q` سره‌خو پدې شرط چې هر راکړیز (Output) یې د یو بل Complement یا سرچپه وي، ۲.۵ انځور د (S-R latch) بلاک دیاګرام ښکاره کوي.
(S-R Latch) د NOR او NAND ګیټ د ترکیب څخه جوړ شوی دی
۳.۵ انځور د (S-R latch) ښودنه د NOR Gate په واسطه
۴.۵ انځور د (S-R latch) ښودنه د NAND Gate په واسطه



S	R	Q	Q'
0	0	1	1 Forbidden
0	1	1	0
1	1	1	0 No change
1	0	0	1
1	1	0	1 No change

۱.۵ جدول د S-R Latch خاصیتونه

هغه S-R latch چې په ۴.۵ انځور کې دی او د NAND gate پواسطه جوړ شوی، په لاندې نقطو کې د S-R latch عملیې روښانه کیږي او په ۱.۵ جدول کې د هرې عملیې مشخصات ښودل شوي.

- که چېرته S=0 او R=0 کړو نو په نتیجه کې Q=Q`=1 لاسته راځي کوم چې

جواز نه لري ځکه چې Q او `Q باید د یو بل سـرچپه وي نو په دې صـورت کې R=S=0 نشي راتلی.

- کـه چیرې S=0 او R=1 په نتیجه کې Q=1 او Q`=0 لکه څنګه چې په ۱.۵ انځور کې ښودل شوي که چېرته S د 0 نه 1 ته بدل شي نو Q نه بدلیږي.

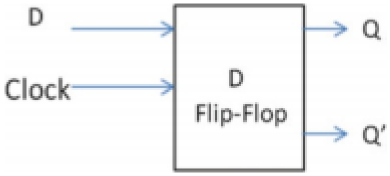
که چېرته S=1 او R=0 شي نو په نتیجه کې Q=0 او Q`=1 لکه څنګه چې په ۱.۵ انځور کې ښودل شوي، که چېرته R له 0 نه 1 ته بدل شي نو په Q کې کوم تغیر نه راځي. نو نتیجه داسې اخلو چې که چیرې S=R=1 نو د Q راکړیز به نه بدلیږی (Q=0 نو Q=0 او که Q=1 وي نو Q=1 پاتې کیږي). دغه S-R latch یو بنیادي سرکېټ دی د D, J-K او T flip-flops له پاره.

۳.۵ ډي فلیپ پلاب (D Flip-Flop)

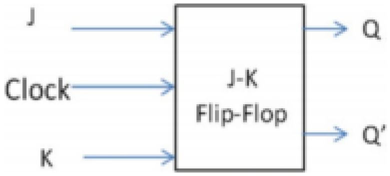
D flip-flop یو 1-bit حـافظه (Memory) ده، او دا د SRAM (Static RAM) او راجستر (Register) د ډیزاین کولو له پاره استفاده کیږي. په ۵.۵ انځور کې د D flip-flop یو Logic Diagram ښـودل شوی. د Flip-Flop ورکړیز واحدات عبارت له D او Clock څخه دي کله چې د Clock قیمت 0 شـی نو S = R = 1 کیږي خو ۲.۵ انځور تـه پـه کتو د Flip-Flop نتیجـه نـه بدلیږي، که د D =0 او د Clock قیمت له 0 څخه 1 ته بدل کړای شي نو په نتیجه کې S = 0 او همدارنګه R = 1 کیږي نو Q = 0 او Q` = 1 لاسـته راځي. معنا دا چې که چېرته D = 0 او Clock ور باندې ولګوو نو د Q راکړیزه (Output) به 0 ته بدلیږي. که د D = 1 او د Clock قیمت له 0 څخه 1 ته بدل کړای شي نو په نتیجه کې S = 0 او همدارنګه R = 1 کیږي. ۲.۵ جدول ته په کتو که چېرته D = 1 کیږي. معنا دا چې که چېرته D = 1 او Clock ور باندې ولګوو نو د Q Output به 1 ته بدلیږي. ۶.۵ انځور کې د D Flip Flop یو بلاک دیاګرام ښـودل شوی او د Clock صعودی لمن " " سره ښودل شوې او په ۲.۵ جدول کې د D Flip Flop مشخصات ښودل شوي دي.

Clock	D	Q
↑	0	0
↑	1	1

۲.۵ جدول د D flip flop مشخصات



۵.۵ انځور: د D flip flop بلاک دیاګرام



۷.۵ انځور J-K flip flop

Clock	J	K	Q
↑	0	0	No change
↑	0	1	0
↑	1	0	1
↑	1	1	Complement

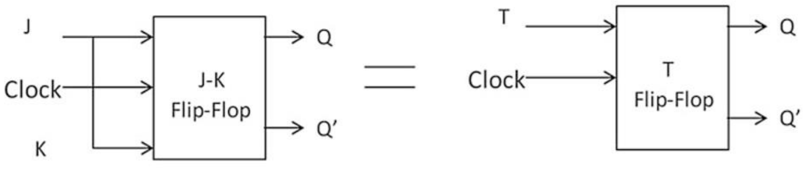
۳.۵ جدول د J-k filp-flop مشخصات

۴.۵ J-K فلیپ پلاپ (J-K Flip-Flop)

۷.۵ انځور کې د J-K Flip Flop یو بلاک دیاګرام ښودل شوی دی چې J-K او Clock د J-K Filp Flop له پاره ورکړیز (Input) دی. د J-K Flip Flop پروګرام (Application) شمیرونکې (Counter) او فریکونسئ تقسیمونکی (Frequency Divider) دی. ۳.۵ انځور کې د J-K Flip Flop مشخصات ښودل شوي او په لاندې نقطو کې د J-K Flip Flop ټول عملیې تشریح کوو.

- که J = K = 0 او Clock Pulse یا زمان سنج پرې ولګوو، نو د Q راکړیز (Output) نه بدلیږي معنا دا چې که چېرته Q = 0 همداسې 0 او که چېرته Q = 1 وی نو همداسې 1 پاتې کیږي.
- که چېرته J = 0, K = 1 او Clock Pulse یا زمان سنج پری ولګوو، نو د Q راکړیز (Output) صفر ته بدلیږي.
- که چېرته J = 1, K = 0 او Clock Pulse یا زمان سنج پری ولګوو، نو د Q راکړیز (Output) 1 ته بدلیږي.

که چېرته J = K = 1 او Clock Pulse یا زمان سنج پری ولګوو، نو د Flip Flop نتیجه او راکړیز (Output) به د اوسنی راکړیز (Output) سرچپه وي، دا پدې معنا چې که جیرې Q = 0 او Clock Pulse یا زمان سنج پری ولګوو، نو راکړیز (Output) به یې 1 ته بدلیږي او که چیرته Q = 1 او Clock Pulse یا زمان سنج پرې ولګوو، نو راکړیز (Output) به یې 0 ته بدلیږي.



۸.۵ انځور د T Flipe-Flop بلاک دیاګرام

Clock	T	Q
↑	0	No change
↑	1	Complement

۴.۵ جدول د T Filpe-Flop مشخصات

۵.۵ ټي فليپ فلاپ (T Flip-Flop)

T Flip-Flop د J-K Flip-Flop یو خاص حالت دی که چېرته په J-K Flip-Flop کې د J او K ورکړیز (Inputs) سره وصل کړو نو په نتیجه کې یو T Flip Flop رامنځته کیږي. ۸.۵ انځور د T Flip Flop یو بلاک دیاګرام ښکاره کوي او د ۴.۵ انځور د T Flip Flop مشخصات ښکاره کوي.
لکه څرنګه چې په ۴.۵ انځور کې ښودل شوی که چیرې T = 0 شي او Clock Pulse یا زمان سنج پرې ولګوو، نو راکړیز (Output) یې نه بدلیږی او که چیري T = 1 شي او Clock پرې ولګوو، نو د Flip Flop راکړیز (Output) به یې د اوسني راکړیز (Output) سرچپه وي.

٦.۵ راجسټر (Register)

D Flip Flop یو 1-bit حافظه (Memory) او یا 1-bit راجسټر (Register) دی. که چېرته د
D Flip-Flops یو ګروپ د یو مشترک Clock څخه استفاده وکړي دې ته راجسټر (Register) وایي. N-bit Register د N په اندازه D Flip-Flops څخه جوړ شوی. که چیري 32 D Flip-Flop د یو مشترک Clock څخه استفاده وکړي نو دغه ته 32-bit Register وایي. ۹.۵ انځور یو 4-bit Register ښکاره کوي، که په دې انځور کې د 1101 په Inputs کې ځای پر ځای کړو او Clock Pulse یا زمان سنج

پرې ولګوو، نو په Output کې به هم 1101 لاسته راشي.

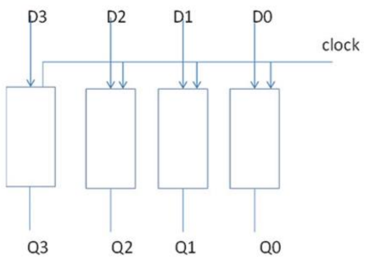
شيفټ راجسټر (Shift Register)

هرشيفټ راجسټر (Shift Register) يو مسلسل ورکړيز (Input) لري د هر کلاک پلس (Clock pulse) له سريال ورکړيز (Input) څخه په راجسټر (Register) کې اچول کيږي او بيا په Register کې هر يو Bit يو Bit مخکې Shift يا بې ځايه کيږي.

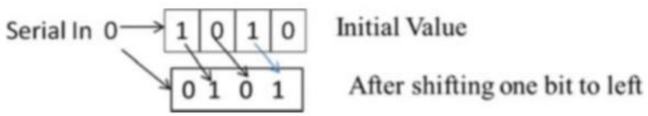
۱۰.۵ انځور يو 4-bit Register چې په ښي طرف بې ځايه(shift) کيږي عمليه راښايي. وروسته د يو bit د بيځايه کېدو څخه د راجسټر (register) محتوا به 0101 وي چې. ۱۱.۵ انځور يو 4-bit serial shift right register راښايي.

مثال: ۱۲.۵ انځور يو 4-bit shift right register راښايي. او همدارنګه د څلورو clock pluses د ځای پر ځای کولو څخه وروسته د راجستر محتوا راښايي، په اول کې داسې فرض کړئ چې د هر D flip flop اولنی Output د صفر سره مساوي وي.

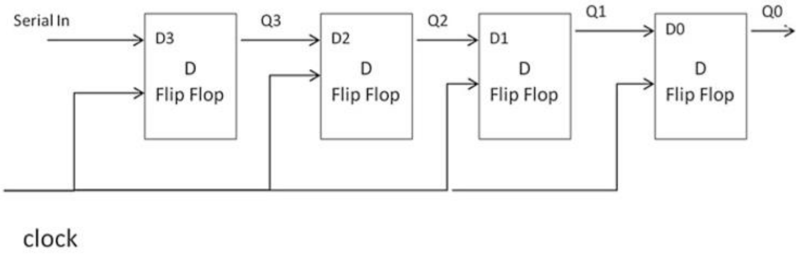
۱۳.۵ انځور يو Shift Left Register د Serial Input سره چې 1110100 دی راښايي، او ۴.۵ انځور کې د Register ټول محتوا د پنځه Clock Pulses د ځای پرځای کولو څخه وروسته راښايي، تاسې داسې په نظر کې ونيسي چې د هر D Flip Flop لومړنی قيمت صفر دی.



۹.۵ انځور ۴- بيټ راجستر



۱۰.۵ انځور 4-bit shift راجستر عملیه



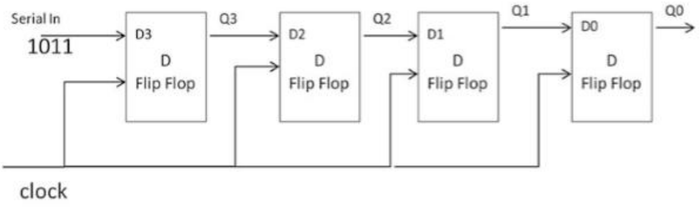
۱۱.۵ انځور 4-bit serial shift right register

چټک ځای پر ځای کېدونکی(Barrel Shifter)

چټک ځای پر ځای کېدوونکی (Barrel Shifter) په راجستر (Register) کې د چپ طرف نه ښی طرف ته د ډېټا دانتقالولو له پاره استعمالېږي، چټک ځای پر ځای کېدوونکي (Barrel shifter) د شیفټ راجستر (Shift Register) په خلاف د ترکېبي منطق (Logic) Combinational څخه استفاده کوي، ترکیبي منطق (Combinational Logic) هیڅ کلاک (Clock) ته ضرورت نلري ځکه خو ډېر تیز (Shifter) ځای پرځای کونکی دی. ۱۴.۵ انځور کې 4-Bit Barrel Shifter او په ۵.۵ جدول کې په چټک ځای پر ځای کېدوونکی (Barrel Shifter) باندی ترسره کېدونکې عملیې راښایي.

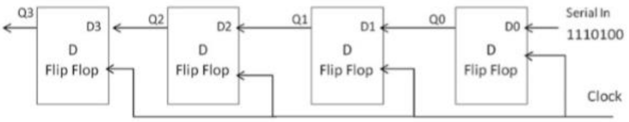
۷.۵ د فریکونســـــي تجزیه کوونکی د J-K Flip-Flop په اســتعمال سره (Frequency Divider Using J-K Flip-Flop)

۱۵.۵ انځور یو J-K Flip Flop چې د فریکونسـي دتجزیه کونکي په ځای کار کوي ښـودل شـوی، د J او K دواړو ورکړیز (Inputs) له 1 سره مساوي او داسې په نظر کې ونیســـئ چې د Q اولنی قیمت د 0 ســـره مســـاوي دی ، لکه څنګه چې په انځور کې ښودل شوی د هر دوه Clock Pulses چې په دغه Flip Flop کې نصب شوی، نو Q یو Clock Pulse رامنځته کوي لکه څنګه چې په ۱۵.۵ انځور کې ښودل شوی، معنا دا چې دغه سرکټ فریکونسي په دوه برخو تقسیم کوي.



Clock#	Q3	Q2	Q1	Q0	
	0	0	0	0	Initial values
1	1	0	0	0	
2	1	1	0	0	
3	0	1	1	0	
4	1	0	1	1	

۱۲.۵ انځور 4-bit Shift Right Register

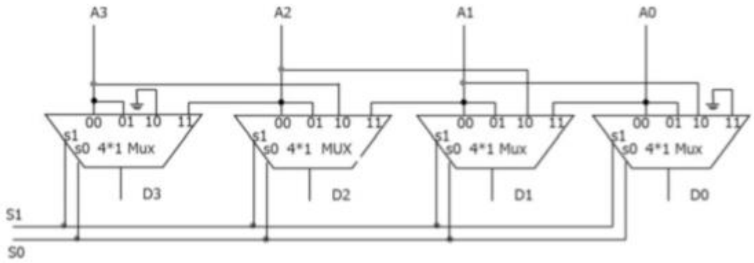


Q3	Q2	Q1	Q0	Clock #	
0	0	0	0	0	initial values
0	0	0	1	1	
0	0	1	1	2	
0	1	1	1	3	
1	1	1	0	4	
1	1	0	1	5	

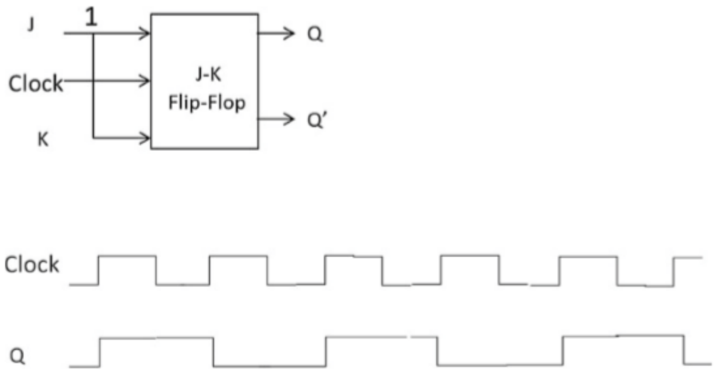
۱۳.۵ انځور 4-bit Shift Right Register

S1	S0	D3	D2	D1	D0
0	X	A3	A2	A1	A0
1	0	0	A3	A2	A1
1	1	A2	A1	A0	0

۵.۵ جدول د چټک ځآی پر ځای کېدوونکي عملیي Barrel shifter operation



۱۴.۵ انځور 4-bit barrel shifter



۱۵.۵ انځور Frequency Divider Using J-K flip-flop

۸.۵ د ترتیبي منطق تحلیل (Analysis Of Sequential Logic)

په ترتیبي منطق کې، د فنکشـــن (Function) مشـــخصــات د صـــدق جدول (Truth Table) پواسـطه ښـودل کیږي، خو د ترتیبي منطق (Sequential Logic) مشـخصـات د سـټیټ ټیبل(State Table) پواسطه ښودل کیږي. State Table درې حـالتونـه لري اوســنی حـالت، راتلونکې حالت او راکړیز (Output). اوســنی حالت یا Present State عبارت دی د Flip Flop د اوسنی حالت یا اوسنی راکړیز (Output) څخـه، او راتلونکې حـالـت عبارت دي د Flip Flop د راکړیز Output څخه چې د ترتیبي منطق (Sequential Logic) پـه ســـر یو Clock ځـای پر ځـای کولو څخـه وروســته لاســته راځي ۱۶.۵ انځور کې یو Sequential Circuit د دوه J-K Flip Flop او دوه حالاتو سره چې A او B دی راښایي.

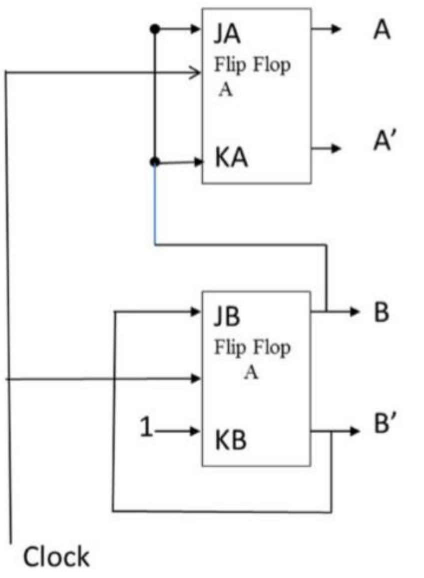
State Table د دوه Columns یا ستون څخه جوړ شـوی یو اوسـنی حالت او بل یي راتلونکې حالت، اوسـنی حالت یا Present State د Flip Flop اوسـنی حالت سره د ټول ممکنـه قیمتونه د A او B له پاره (00, 01, 10, 11) دي چې ۶.۵ دول ښـــکاره کوي.

په ۶.۵ جدول کې تاســې لومړی قطار یا Row په نظر کې ونیســئ، که د اوســنی حالت یا Present State قیمت 00 وو (معنا دا چې $A = 0$ او $B = 0$)، نو په راتلونکې حالت یا Next State کې د Sequential Logic په ســـر د Clock د نصب کولو څخه د Flip Flop نتیجه یا Output لاسته راځي.

که $A = 0$ ، $B = 0$ و او په نتیجه کې یي $JA = KA = 0$ او $JB = KB = 1$ سره شي او د Clock Pulse د ځای پر ځای کولو څخه وروسته د Flip Flop په نتیجه کې $A = 0$ او B=1 لاسته راځي او همدارنگه په Next State کې $AB = 01$ په لاس راځي.

په ۶.۵ جدول کې تاسې دوهم قطار یا Row په نظر کې ونیسئ، که د اوسني حالت یا Present State قیمت 01 وو (معنا دا چې $A = 0$ او $B = 1$)، نو په نتیجه

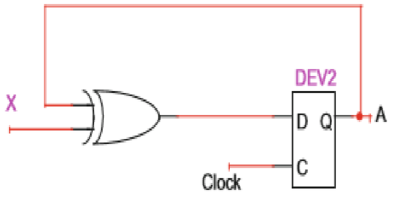
کې يي JA = KA = 1 او JB =0 او KB = 1 سره شي او د Clock Pulse د ځای پر ځای کولو څخه وروسته د Flip Flop په نتیجه کې A = 1 او B = 0 لاسته راځي او همدارنګه په Next State کې AB = 10 په لاس راځي، او همداسې د نورو قطارونو له پاره هم.



۱٦.۵ انځور ترتیبي منطق Sequential Logic

Present state		Next state	
A	B	A	B
0	0	0	1
0	1	1	0
1	0	1	1
1	1	0	0

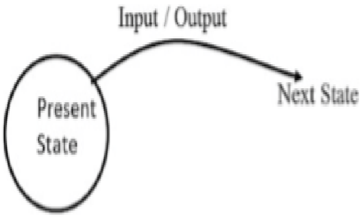
٦.۵ جدول د ۱٦.۵ انځور له پاره State Table راښایي



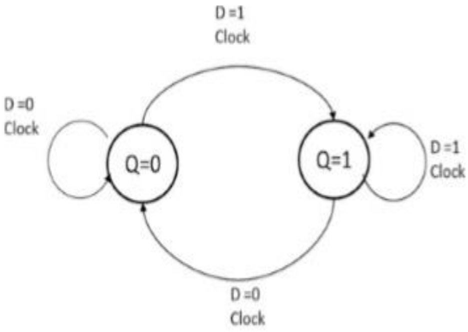
۱۷.۵ انځور مسلسل (Sequential) سرکټ

Present state	Next state for X = 0	Next state for X = 1
A	A	A
0	0	1
1	1	0

۷.۵ جدول د ۱۷.۵ انځور له پاره *State table*



۱۸.۵ انځور *State Diagram*



۱۹.۵ انځور د *Flip-Flop* D له پاره *State diagram*

مثال: د ۱۷.۵ انځور له پاره State Diagram جوړ کړئ؟

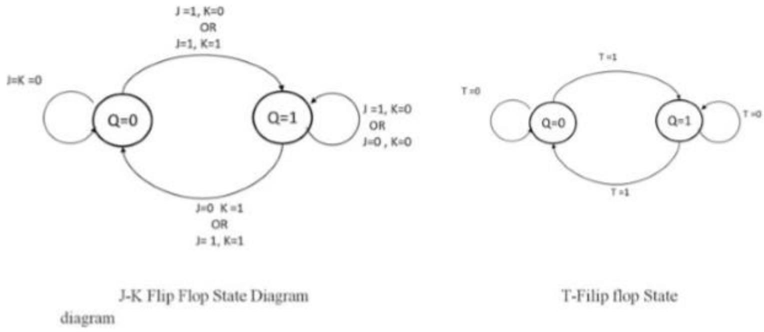
۱۷.۵ انځور یو خـارجي (External) د X ورکړیز (Input) لري او ۷.۵ جـدول دوه سـتونونه د راتلونکې حالت له پاره ښـکاره کوي چې د $X = 0$ او بل د $X = 1$ له پاره دی.

۹.۵ سټیټ ډیاګرام (State Diagram)

د Sequential Logic د مشـخصـاتو د ښـودلو له پاره بله لاره د State Diagram دی، لکه څنګه چې په ۱۸.۵ انځور کې ښـودل شوی. Present State یې د Flip Flop اوسنی قیمت دی او د Clock Pulse د ځای پر ځای کولو څخه وروسته دغه Present State اوسنی حالت په Next State راتلونکې حالت باندی بدلیږي.

D Flip Flop State Diagram

په ۱۹.۵ انځور کې د D Flip Flop یو State Diagram ښودل شوی او په لاندې ډول یې تشریح کوو:



۲۰.۵ انځور J-K And T flip-flops State Diagrams

Q(t)	Q(t + 1)	D
0	0	0
0	1	1
1	0	0
1	1	1

۸.۵ جدول Excitation Table For D flip-flop

1- که چېرته Q = 0 (Present State) د D = 1 ورکولو سره او همدارنګه د Clock د ځای پر ځای کولو نه وروسته د Q قیمت له 0 څخه 1 ته بدلیږي (Next State)

2- که Q = 1 شي او D = 1 او Clock پرې ځای پر ځای کړو، نو Output یې 1 پاتې کیږي.

3- که Q = 1 شي او D = 0 او Clock پرې ځای پر ځای کړو، نو Output یې 0 ته بدلیږي.

4- که Q = 0 شي او D = 0 او Clock پرې ځای پر ځای کړو، نو Output یې 0 پاتې کیږي.

۲۰.۵ انځور کې د J-K او T Flip Flop له پاره State Diagram ښودل شوي.

۱۰.۵ فلیپ فلاپ متحرک جدول (Flip-Flop Excitation Table)

متحرک جدول د فلیپ فلاپ (Flip-Flop) د راکړیز یا ورکړیز کارول څرګندوي ددی له پاره چې له مخکې څخه تعین شوی راکړیز ترلاسه کړو.

ډي فلیپ فلاپ متحرک جدول(D Flip-Flop Excitation Table):

۸.۵ جدول موږ ته د D Flip-Flop له پاره یو Excitation Table راښايي.
په جدول کې Q(T) اوسنی حالت او Q(T+1) راتلونکې حالت راښايي.

د *Excitation Table* عملیې: د ۸.۵ جدول اول ستون په نظر کې ونیسئ که د Q(T)=0 (اوسنی حالت) او د Clock Pulse د ځای پر ځای کولو څخه وروسته وي او د Q(T+1) راکړیز (Output) بیا هم 0 پاتې وي نو D باید 0 وي.
د ۸.۵ جدول دوهم ستون په نظر کې ونیسئ که چېرته د D Flip-Flop اوسنی حالت 0 وي، او د راتلونکی حالت یا Q(T+1) نتیجه 1 وي نو د D له پاره Input ورکړیز باید 1 اوسي.

Q(t)	Q(t + 1)	J	K
0	0	0	d
0	1	1	d
1	0	d	1
1	1	d	0

۹.۵ جدول د J-K flip-flop Excitation Table

Q(t)	Q(t + 1)	T
0	0	0
0	1	1
1	0	1
1	1	0

۱۰.۵ جدول د T flip-flop Excitation Table

د ۸.۵ جدول دریم ستون په نظر کې ونیسئ که اوسنی حالت 1 وي، او د راتلونکی حالت نتیجه 0 ته تقرب وکړي نو د D-Input باید مساوي وي له 0 سره.
د ۸.۵ جدول څلورم ستون په نظر کې ونیسئ که اوسنی حالت 1 وي، او د راتلونکي حالت نتیجه هم 1 پاتی شي نو د D-Input باید مساوي وي له 1 سره.

:J-K Flip-Flop Excitation Table

په ۹.۵ جدول کې د J-K Flip-Flop Excitation Table ښود ل شوی دی، او په لاندې ډول یې واضح کوو:

۱. د ۹.۵ جدول اول ستون په نظر کې ونیسئ که د Flip-Flop اوسنی حالت 0 وي، او د Clock Pulse د ځای پر ځای کولو څخه وروسته د Q(T+1) راکړیز (Output) بیا هم 0 پاتې وي نو J باید 0 وي. او K کېداشي 0 یا 1 شي
۲. د ۹.۵ جدول دوهم ستون په نظر کې ونیسئ که د Flip-Flop اوسنی حالت 0 وي، او د Clock Pulse د ځای پر ځای کولو څخه وروسته د Q(T+1) راکړیز (Output) بدل شي په 1 باندی نو J باید 1 وي. او K کېداشی 0 یا 1 شي.
۳. د ۹.۵ جدول دریم ستون په نظر کې ونیسئ، که د Flip-Flop اوسنی حالت 1 وي، او د Q(T+1) راکړیز (Output) بدل شي په 0 باندی نو K باید 1 وي. او J کېدای شي 0 یا 1 شي.
۴. د ۹.۵ جدول څلورم ستون په نظر کې ونیسئ، که د Flip-Flop اوسنی حالت 1 وي، او د Q(T+1) راکړیز (Output) بیا هم 1 پاتې شي نو K باید 0 وي. او J کېداشی 0 یا 1 شي.

:T Flip-Flop Excitation Table

په لاندې جدول کې د T Flip-Flop Excitation Table ښودل کیږي.

۱۱.۵ شمیرونکی یا Counter

شـــمیرونکی یا Counter یو ترتیبي منطق دی چې په یو ســـرکټ پورې د ټولو وصل شوو Pulses یا ضربان ورکونکو شـــــمیرلو له پاره استفاده کـــیږي او همدارنگه د Clock د فریکونســــــي د ویشلو له پاره که ســیــــستم د Clock 16 Hz لرونکی وي دا امکان لري چې د یو شـــمیرونکې یا Counter په اســـتفادې ســـره دغه 16 Hz په 4 Hz بدل کړو.

په لاندې نقطو کې ښودل شوي چې څنگه یو شمیرونکې یا Counter ډیزاین کړو:

Present state		Next state	
A	B	A	B
0	0	0	1
0	1	1	0
1	0	1	1
1	1	0	0

۱۱.۵ جدول د شمیرونکي جدول State Table(Counter)

Present state		Next state					
A	B	A	B	JA	KA	JB	KB
0	0	0	1	0	d	1	d
0	1	1	0	1	d	d	1
1	0	1	1	d	0	1	d
1	1	0	0	d	1	d	1

۱۲.۵ جدول د شمیرونکي له پاره Excitation Table

a. یو داسې سلسله معرفی کړئ چې شمیرونکې یې شمیر کړي.
b. د دغه سلسلې څخه په استفادې د Flip-Flop نمبر معلوم کړئ.
c. د Flip-Flop نوعه معلومه کړئ.
d. دپورته جوړی شـوي ســلســلې په مرســته د Flip-Flop له پاره یو State Table

جوړ کړئ.

e. د دغـه State Table او Flip-Flop Excitation Table پـه واســـطـه دشميرونکي يا Counter له پاره يو Excitation Table جوړ کړئ.

f. د K-Map څخه په استفادې د هر Flip-Flop له پاره د ورکړيز (Input) ميتودونه اونتيجی په لاس واوړئ.

g. او په اخيره کې د شميرونکی يا Counter يو ترتيبي منطق رسم کړئ.

مثال: يو شميرونکی د 3 – 2 – 1 – 0 د شميرلو له پاره کړی چې د J-K Flip-Flop پواسطه تکراريري. په شميرونکې کې تر ټولو غټ نمبر 3 دی کوم چې د باينری (Binary) 11 پواسطه ښودل کيږي. نو دوه Flip-Flop ته ضرورت دی او هغه په A او B نوموو لکه څنګه چې په ۲۲.۵ انځور کې ښودل شوی. او په ۱۱.۵ جدول کې د شميرونکی يا Counter له پاره State Table ښودل شوی.

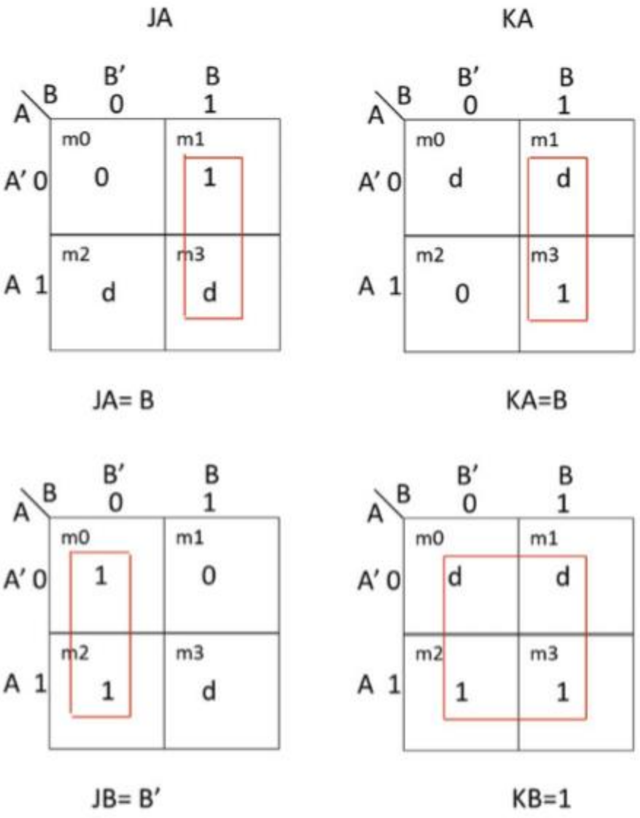
اوســنی حالـت يي د Flip-Flop اوسـنې نتيجه، او راتلـونکې حالت يي د Flip-Flop نتيجه د Clock Pulse د ځای پر ځای کولو څخه وروسته راښايي.

۱۲.۵ جدول کې د J-K Flip-Flop Excitation Table پواسـطه جوړشوی د شميرونکي يا Counter له پاره د Excitation Table ښودل شوی.

اول ســتون په نظر کې ونيســـــي، د J-K Flip-Flop اوســنی نتيجه 00 ده يعنې $(A = 0, B = 0)$ ، او ضــرورت دا ده چې راکړيز (Output) بايد په 01 بدل شــي يعنې ($A = 0$ او $B = 1$ ته)، نو بايد JA=0, KA=D (Don’t Care) شــي d فرق نه کوي چې 0 شــي يا 1، د A له نظره چې بايد 0 پاتې شــي، اما د B له نظره چې بايد $JB = 1$ او KB = d (don’t care) شــي d يعنی فرق نه کوي چې 0 شي يا 1، ددی له پاره چې د B نتيجه د 0 څخه په 1 بدله شي.

دوهم ســتون په نظر کې ونيســئ، د J-K Flip-Flop اوســنی نتيجه 01 ده يعنې (A=0, B=1)، او تقاظا داده چې نـتيـجه بايـد 10 شــي يـعـنــي (A =1 And B =0) نــو بــايــد JA = 1, KA = d او JB = 0, KB = d شي.

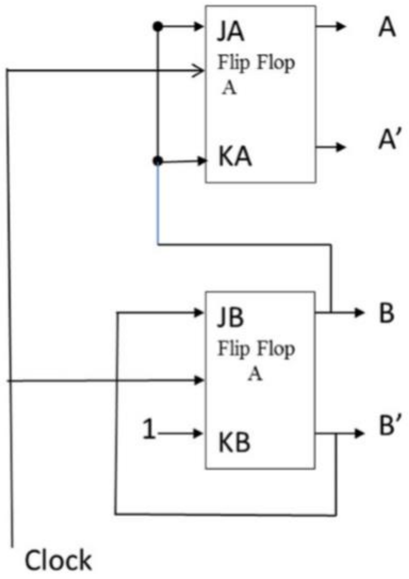
تقاضـــا داده ده چې بـایـد د Flip-Flop لـه پـاره ورکړیز میتودونـه (Input Functions) پیدا کړل شي،که اوسنی حالت د ورکړیز (Inputs) په توګه او JB ,KA, JA او KB د ۱۲.۵ جـدول راکړیز (Outputs) وي او دغـه Outputs څخه د K-maps د جوړولو له پاره استعمالیږي، او د ۱۲.۵ جدول په لوسـتلو سره او همدارنګه د ۲۱.۵ انځور د A, KA, JB او Kb له پاره K-maps راښايي. ورکړیز میتودونه د Flip-Flop له پاره عبارت دي له JA = JB = B, JB =B` او KB = 1 او په ۲۲.۵ انځور کې د 2-bit شمیرونکې سرکټ بنودل شوي.



۲۱.۵ انځور k-Maps د 2-bit شمیرونکې له پاره

۱۲.۵ خلاصه

- ترتیبي منطقي سرکټ Clock یا ساعت ته ضرورت لري چې عملیي ترسره کړي.
- S-R Latch د Flip-Flop اساسي برخه ده.
- S-R Latch کولای شــو د ننډ (NAND) او نار (NOR) ګیټونو (Gates) پواسطه جوړ کړو.
- د ترتیبي منطق اساسي واحد Flip-Flop دی.
- Flip-Flop د دوه راکړیزو (Outputs) Q او `Q ســره د حافظې (Memory) یو عنصر دی
- په D Flip-Flop پورې د Clock په ترڅلو ســره د D له پاره د ورکړیز (Input) قیمت د Q له پاره د راکړیز (Output) قیمت ته کاپي کیږي.
- D Flip-Flop د راجسټر (Register) د ډیزاین کولو له پاره استفاده کیږي.



۲۲.۵ انځور د 2-bit شمیرونکي له پاره ترتیبي منطق (Sequential Logic)

- راجستر (Register) د D Flip-Flop يو داسې مجموعه ده چې يو Clock سره شريکوي.
- J-K Flip-Flop د Counter يا شميرونکې د ډيزاين کولو له پاره استفاده کيږي.
- د J-K Flip-Flop د ورکړيز (Inputs) د يو بل سره د وصل کولو څخه T Flip-Flop لاسته راځي.
- Sate Table او State دياګرام د يو ترتيبي سرکټ عمليي راښايي.

په شپږم فصل کې د کپيوټر د جوړښت څخه بحث شوی دی او لاندې موارد پکې رانغاړل شوي دي: د Microcomputer اساسي اجزاوې، د پروسيس مرکزي واحد (CPU) مختلفی تکنالوژي، د پروسيس مرکزي واحد (CPU) جوړښتونه، څوييز کور پروسيسر (Multicore Processor)، د کمپيوټري هداياتو د اجرا کېدو مرحلې،پايپ لينينګ (Pipelining)، او د د کوچني کمپيوټر وصلوونکي (Microcomputer Buses).

پوښتنې

۱. سوال: لاندې جدول د D Flip-Flop له پاره پوره کړئ.

D	Q(*t*) present output	Q(*t* + 1) next output
0	0	
0	1	
1	0	
1	1	

۲. سوال: لاندې جدول د J-K Flip-Flop له پاره پوره کړئ.

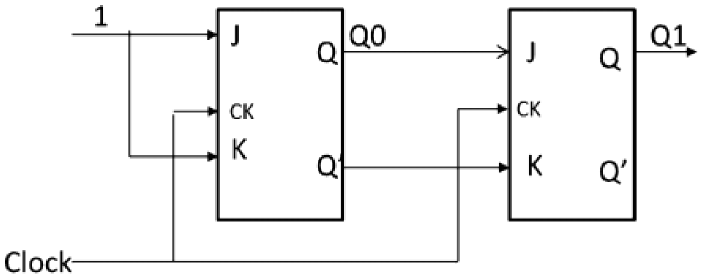
J	K	Q(*t*) present output	Q(*t* + 1) next output
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

۳. سوال: لاندې جدول د T Flip-Flop له پاره پوره کړئ.

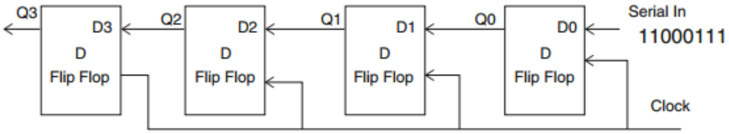
T	Q(*t*) present output	Q(*t* + 1) next output
0	0	
0	1	
1	0	
1	1	

۴. سوال. په لاندې انځور کې یو ترتیبي منطق ښودل شوی، لاندې جدول پوره کړئ په داسې حال کې چې د $Q1 = 0$ او د $Q2 = 0$ ابتدایي قیمتونه وي. د Logisim سافټویر څخه جواب د تایید له پاره استفاده وکړئ.

Clock	Q0	Q1
Initial value	0	0
Clock #1		
Clock #2		
Clock #3		

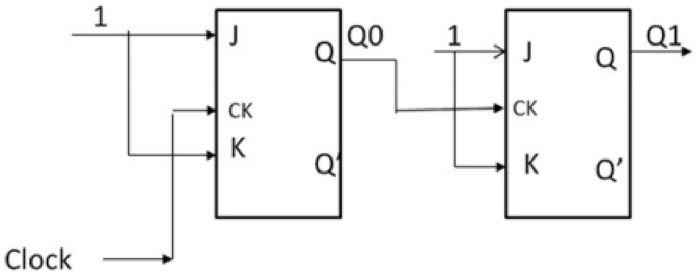


۵. سوال. يو 8-bit لرونکي راجسټر (Register) د D Flip-Flop پواسطه وښايئ؟



۶. سـوال. لاندې يوشـيفټ راجسـټر (Shift Register) در کړل شـوی، د پنځه ځلې Clock Pulses د تېرلو وروسته يې نتيجه په لاس راوړئ؟

۷. سـوال: د لاندې ترتيبي منطق درکړل شـوی، د $Q0 = 0$ او $Q1 = 0$ ابتدايي قيمتونه په نظر کې ونيسـئ او Flip-Flop خپل حالت د Clock Pulse په پورته کېدونکي څنډه کې بدلوی، لاندې جدول پوره او بيا خپل جواب په Logisim کې ميزان کړئ.



Clock	Q0	Q1
Initial value	0	0
Clock #1		
Clock #2		
Clock #4		
Clock #4		

۸. سوال : لاندې Excitation جدول د J-K Flip-Flop له پاره پوره کړئ؟

Q(t)	Q(t + 1)	J	K
0	0		
0	1		
1	0		
1	1		

۹. سـوال: يو شـميرونکی يا Counter ډيزاين کړئ چې 7-6-5-4-3-2-1-0 اعداد وشميري او تکرار هم شي.

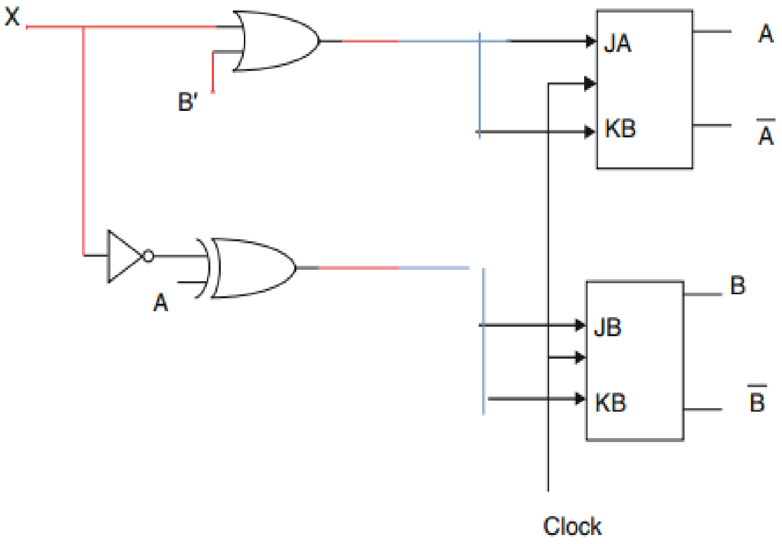
- د J-K Flip-Flop څخه استفاده وکړئ.
- د T Flip-Flop څخه استفاده وکړئ.
- خپل ډيزاين په Logisim کې تاييد کړئ.

۱۰. سوال: لاندې State جدول له پاره State دياگرام پيداکړئ؟

	AB	AB
AB	X = 0	X = 1
00	01	10
01	10	00
10	11	01
11	00	10

۱۱. سوال: د لاندې راجستر (Register) په محتوا کې به د پنځه خلې چپ طرف ته د شفټ (Shift) کولو وروسته څه وي؟

۱۲. سوال: د لاندې سرکټ له پاره State جدول او State ډیاګرام وښایئ؟



شپږم څپرکی

د کمپیوټر د جوړښت پیژندګلوي

(Introduction To Computer Architecture)

موخې: د دې څپرکي په په ختمیدو سره به تاسې وکولای شئ چی لاندی کارونه تر سره کړئ

- د کوچنې کمپیوټر (Microcomputer) د جوړښتیزو برخو پیژندنه.
- د پروسیس مرکزي واحد (CPU) د جوړښتیزو برخو پیژندنه.
- د پروسیس مرکزي واحد (CPU) د بیلا بیلو جوړښتیزو ټکنالوژیو تر منځ توپیرونه پیژندل.
- د څو هستو لرونکي پروسیسر (Multi-Core Processor) د جوړښت پیژندنه.
- د Risc (کوچنی کمپیوټر خو د مطلوبو لا رښوونو سره) پروسیسر پرتله له Cisc (د پیچلو لارښوونو سیټ، کمپیوټر) پروسیسر سره.
- د وان نیومن (Von Neumann) او هاروارډ (Harvard) د جوړښـتونو تر منځ توپیرونه.
- د 32 او 64بیټ (bit) پروسیسرونو تر منځ توپیرونه پیژندل.

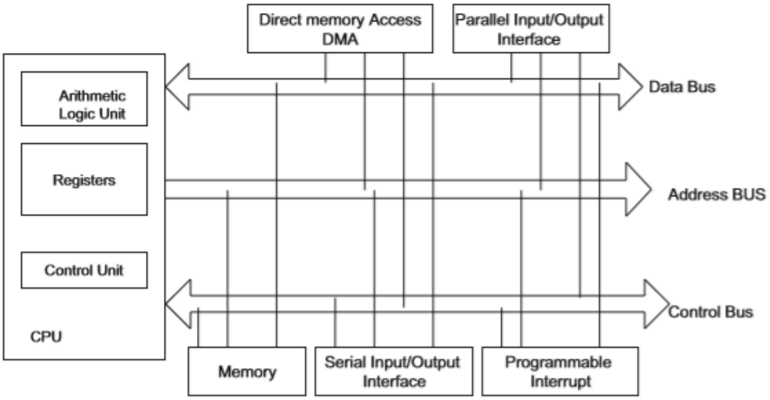
- د یوې لارښوونې (Instruction) د تر سره کېدلو د مراحلو سره بلدتیا.
- د لارښوونو د پایپ لاین کېدلو (Instruction Pipelining) د ګټې پیژندل.
- د کوچنی کمپیوټر (Micro-Computer) د بیلا بیلو وصلونکو (Buses) پیژندګلوي د عمومي وصلونکې(USB Bus) عملیه تشریح کړئ.

۱.٦ پیژندګلوي

لکه څرنګه چې د یو تعمیر او کور جوړښت د نوموړی تعمیر جوړښت او کړنې معرفي کوي همدارنګه د یو کمپیوټر جوړښت هم د کمپیوټر جوړښت او کړنې راپیژني. د هر کمپیوټري سیستم بیلا بیلې برخې په داسې ډول سره نقشه شوې دي چې یو له بل سره اړیکه نیسي او بیلا بیلې کړنې تر سره کوي.

۲.٦ د کوچني کمپیوټر جوړښتیزې برخې (Components Of (Microcomputer

هر معیاري کوچنی کمپیوټر له ځینو برخو لکه د پروسیس مرکزي واحد (CPU)، وصلوونکو (Buses)، حافظې (Memory)، موازي راکړیز او ورکړیز واحدونو (Parallel I/O)، مسلسل راکړیز او ورکړیز واحدونو (Serial I/O)، پروګرام کېدونکي راکړه یز او ورکړه یز مزاحمتونه (Programmable I/O Interrupts) او په مستقیم ډول سره د حافظې د لاسرسي برخې (DMA) څخه جوړ شوی دی چې په لاندې شکل کې یې په ښکاره ډول سره کتلای شئ.



۱.٦ انځور د کوچني کمپیوټر(Microcomputer) جوړښتیزې برخې

۱.۲.٦ د پروســیس مرکزي واحـد (CPU): نوموړې برخې ته د کمپیوټر مغز هم وایي چې د راکړیزو الو(Devices Input) څخه ډیټا تر لاسه کوي، نوموړې ډیټا په معلوماتو بدلوي ، دغه معلومات زیرمه کوي او ورکړیزو الو (Output Devices) ته نوموړي معلومات انتقالوي .

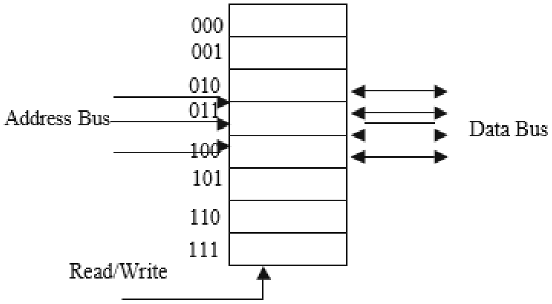
پروسیس مرکزي واحد (CPU) په لاندې دریو لویو برخو باندې ویشل شوې ده:

۱. حسابي او منطقي برخه (Arithmetic Logic Unit)
۲. اداره کوونکې برخه (Control Unit)
۳. راجسټرونه (Registers)

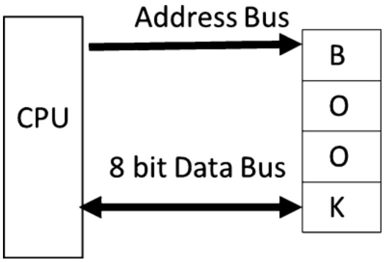
د حســابي او منطقي برخې (ALU) دنده دا ده چې حســابې کړنې لکه جمع، تفریق، ضرب او تقسیم او همدرانګه منطقې کړنې لکه انډ(AND)،اور(OR) او ناټ (NOT) تر سره کړي. کنټرونکي برخه (CU) دنده دا ده چې راکړیزی او ورکړیزی آلی (Input/output Devices) اداره کړي ، د کمپیوټر د نورو برخو لـه پاره اداره کوونکي سیګنلونه لکه د لوستلو او لیکلو سیګنل تولید کړی او د لارښوونو اجرا ت تر سره کوي.

راجسترونه چې په کمپیوټري سیستم کې تر ټولو زیات سرعت لرونکې زېرمه ده، چې له حافظې (Memory) څخه د حسابي او منطقي برخې (ALU) له پاره ډېټا تر لاسه کوي.

۲.۲.٦ د پروسیس مرکزي واحد وصلوونکې (CPU Buses): کله چې له یو څخه زیات لاینونه یو ډول معلومات انتقال کړي د وصلونکې (Bus) په نوم سره یادېږي. د کوچني کمپیوټری سیستم په منځ کې تر ټولو مشهور وصلونکې (Bus) عبارت دي له: د وصلولو ادرس (Address Bus) ، د وصلولو کنټرولونکی (Control Bus) او ډېټا وصلونکي (Data Bus) څخه.



۲.٦ انځور میموری د درې ادرس او څلور د ډېټا لاینونو سره



۳.٦ انځور د معلوماتو وصلونکې د ۸ بیټه سي پي یو سره (CPU) سره

ادرس وصلونکې (Address Bus): د حافظې (Memory) په مدغم شوي سرکټ (IC) کې د هغه موقعیتونو شمیر چې د آدرس کېدلو وړتیا ولري د آدرس په واسطه مشخص کېږي چې د 2^n په مرست پیدا کېږي چې n د آدرس لاینونو مجموعه ښکاره کوي. د بیلګې په توګه که چېرې زموږ د کمپیوټر ادرس وصلونکې (Address Bus) د درېو لاینونو درلودونکې وي نو په حافظه (Memory) کې به یې 2^3 د آدرس کېدلو وړ موقعیتونه شتون لري. لکه څرنګه چې په لاندې شکل کې لیدلای شی د یو آدرس بس د لاینونو مجموعه په مستقیم ډول سره په میموري کې د آدرس کېدلو وړ موقعیتونو سره اړیکه لري. یعنې کـه چېرته د آدرس وصلونکې (Address Bus) د لاینونو مجموعه 32وي نو د آدرس کېدلو وړ موقعیتونو اندازه به مو په حافظه (Memory) کې 2^{32} وي.

ډیټا وصولونکی (Data Bus): د دې له پاره استفاده کېږي چې په حافظه (Memory) کې ډیټا له یو ځای څخه بل ځای ته انتقال کړي. په ۳.٦ شکل کې د حافظې (Memory) هر موقعیت کولای شي چې ۴ بیټه معلومات زیرمه کړي. چې په مجموعي ډول سره ۸* ۴ بیټه یعنې ۳۲ بیټه معلومات زیرمه کولای شي چې د حافظې (Memory) د مدغم شوي سرکټ (IC) اندازه کیږي چې د $2^n * m$ فورمول له مخې یي پیدا کړو په داسې ډول سره چې n د ادرس وصلونکې (Address bus) د لاینونو مجموعه او m د حافظې (Memory) د یو موقعیت د زیرمه کولو وړتیا ښکاره کوي. د یادونې وړ ده چې د ډېټا وصلونکي (Data Bus) اندازه د مرکزي پروسس کوونکی واحد (CPU) د کار کولو په څرنګوالي کې خورا مهم رول لوبوي. اوسني مرکزي پروسس کوونکي واحدونه (CPUs) د $32bits$ او $64bits$ ډېټا وصلونکو لرونکې ده یعنی کولای شي چې له حافظې (Memory) څخه په یو وخت کې $32bits$ یا $64bit$ ډېټا ولولي او یا یې هم ورته واستوي. په پخوا وختونو کې د مرکزي پروسیس کوونکی واحد (CPU) $8-bits$ ډېټا بس استفاده کولو یعنی په یو وخت کې کولای شول چې له حافظې (Memeory) څخه $8-bits$ ډېټا تر لاسه کړي او یا هم حافظې

(Memory) ته واستوي. د بيلګې په توګه که چېرته موږ غواړو چې د $(Book)$ لغت په حافظه (Memory) کې وســاتو نو اړينه ده چې حافظې ته ۴ ځله رســيدنه (Access) وکړو چې په هر ځل لاسرسـي (Access) سـره کولای شـو يوازې او يوازې ۸ بيټه چې يو بايـټ $(Byte)$ کېږي ترلاســه (Access) کړو. اما کـه چېرته د معلوماتو وصـــلونکي $(Data\ bus)$اندزه له ۸ بيټو څخه ۳۲ بيټو ته زياته کړو کولای شـو چې نوموړی لغت د حافظې (Memory) څخه په يو ځل لاسرسـي(Access) سـره تر لاسـه کړو او يا يې هم په حافظې (Memory) کې وساتو.

د کنټرول وصـــلوونکی (Control Bus): دا د کمپيوټر نورې برخې کنټرول او اداره کړي لـه کنټرول واحـد (Control Unit) څخـه اداره کوونکي ســګنلونـه اخلي او د کمپيوټر نورو برخو تـه يې انتقالوي او همدارنګه ځينې وخت نوموړي اداري ســګنلونه د کمپيوټر له نورو برخو څخه هم اخلي.

ځينې مشهور اداره کوونکی سګنلونه په لاندې ډول سره دي:

۱. **د لوســتلو سيګنل (Read Signal)**: د حافظې (Memory) آدرسونو او همدارنګه د ورکړيزو او راکړيزو آلو (Input/output Devices) څخـه د ډيټـا د لوســتلو لـه پـاره استفاده کېږي.
۲. **د ليکلو ســيګنـل (Write Signal)**: د ليکلو لـه لاين څخـه پـه اســتفادې د حافظې (Memory) په يو آدرس کې ډيټا ساتي.
۳. **مزاحمت (Interrupt)**: يو مزاحمتی سيګنل منځ ته راوړي.
۴. **د وصـلونکې غوښـتنه (Bus Request)**: يوه آله يې د وصلونکي(Bus) د غوښتلو له پاره استفاده کوي.
۵. **د وصـلونکې اجازه (Bus Grant)**: د دې سـيګنل په مرسـته يو غوښـتونکې آلې ته د وصلونکې (Bus) د استفاده کولو اجازه ورکول کېږي.
۶. **راکړيز او ورکړيز ليکلو او لوستلو سيګنل**: په مرسته يې کولای شو چې ورکړيز يا راکړيز الې (Input/output Devices) ته ډيټا واستوو او يا يې هم ورڅخه تر لاسه کړو.

۳.۲.۶ د ۳۲ بیټه او ۶۴ بیټه پروسیس مرکزي واحد ترمنځ توپیر

(Bit Versus 64-Bit CPU-۳۲): د راجســټرونو اندازه د د مرکزي پروســیس کوونکی واحد (CPU) د کار کولو په څرنګوالي کې ډېر مهم رول لوبوي۳۲ بیټه (bit) پروسـیس مرکزي واحد (CPU) په ۳۲ بیټه (bit) معلوماتو باندې عملیه تر ســره کوي چې د نوموړې مرکزي پروســیس کوونکی (CPU) د ریجیســټرونو اندازه هم۳۲ بیټه (bit) ده او همدارنګه حســابي او منطقي برخه (ALU) یې هم کولای شي په ډېټا باندې ۳۲ بیټه (bit) عملیه تر سره کړي.

۶۴ بیټه (bit) پروســیس مرکزي واحد(CPU) په ۶۴ بیټه (bit) معلوماتو باندې عملیه تر سـره کړي چې د نوموړې مرکزي پروسـیس کوونکی (CPU) د ریجیسټرونو اندازه هـم ۶۴ بیټه (bit) ده او همدارنګه حسابي او منطقي برخه (ALU) یې هم کولای شي په ۶۴ بیټه (bit) معلوماتو باندې عملیه تر سره کوي.

اکثره میزي کمپیوټرونـه (Desktop Computer) او ســرور(Server) کمپیوټرونـه د اینټل پروســیســر Intel Processor او پرمختللي کوچني آلو(Advanced Micro Devices) پروسـیسـرونه اسـتفاده کوي چې نوموړی پروسـیسـر دوه ډولونه یعنې ۳۲ بیټه (bit) او۶۴ بیټـه (bit) دواړه لري او نوموړي ډولونـه یعنې Intel او پرمختللي کوچني الې (Advanced Micro Devices) د یو شان جوړښت لرونکي دي او هغه پروګرام چې په یو پروسـیسـر باندې چلول (Run) کېدلای شـي کولای شـو په بل یې هم چلولي شـو یعنی (Run) کولی شو.

۳.۶ د مرکزي پروسېس کوونکی واحد ټکنالوژي (CPU (Technology

پـه عمومي ډول ســره د CPU لـه پاره دوه ډوله تکنالوژې اســتفاده کیږي چې عبـارت دي لـه د پیچلي کمپیوټري لارښــوونې ســیــټ (CISC) او په یو دوران کې یوه لارښونه اجرا کول (RIS) څخه.

د پیچلي کمپیوټري لارښوونې سیټ (Computer (Complex Instruction Set یا (CISC) په 1976م کال کې د اینټل (Intel) کمپنۍ یو کوچنی پروســیســر چپ د 8086 په نوم ســره منځ ته راوړو چې کولای یې شــول یوازې په ۱٦ بیټه (bit) معلوماتو بـاندې عملیې تر ســره کړي چې د اعشــاري عددونو د عملیو له پاره یې کومې ځانګړې لارښــوونې نه درلودې. مګر نني پینټیم (Pentium) پروسـیسـرونه کولای شـي چې تر۳۲ بیټه (bit) او حتي تر ٦۴ بیټه (bit) پورې معلومات پروسـیس کړي او همدارنګه اعشاري عملیې هم تر ســره کولای شـــي. د یـادولو وړ ده چې اینټـل (Intel) اوســـني پینټیم (Pentium) پروسـیسرونه په داسې ډول سره ډیزاین شوی دD چې کولای شي د پخوانیو 80x86 پروسـیسرونو لارښوونو (Instructions) باندې هم عملیې تر سره کړای شي .

د CSIS پروســیســرونو بـل ډول د (VAX 11/780) پـه نوم ســره یـادیږي چې د (PDP − 11) کمپیوټرونو لـه پـاره اســتفـاده کیږي. د یـادونې وړ ده چې د نوموړې ټکنالوژی (CSIS) لرونکي پروســیســرونه بیلا بیلې لارښــوونې د بیلا بیلو آدرســونو له ســبکونو(modes) ســره لري د بیلګې په توګه (VAX 11/780)کولای شـي چې د 300 بیلا بیلې لارښــوونې (Instruction) د ۱٦ مختلفو آدرس طریقو(modes) پـه درلودلو سره تر سره کړي.

د پیچلیو کمپیوټري لارښــوونو (CISC) ځینې ځانګړنې کولای شــو په لاندې ډول ســره لنډیز کړو:

۱. د لارښوونو (Instructions) زیات تعداد لرل.
۲. ډیر د آدرس طریقې (Addressing Modes) لري.
۳. د لارښوونو (Instructions) اندازه یې یو له بل سره توپیر لري.
۴. کنټرول واحد (Control Unit) یې پروګرام شوی دی.
۵. ډیـری لارښـــوونـې کـولـی شـــي پـه حـافظـه (Memory) کې عمـل کوونکي (Operands) اداره کړي.

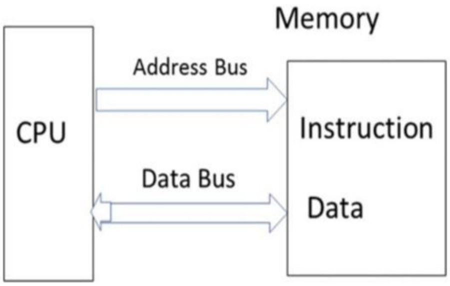
په یوه دوران کې یوه لارښـونه اجراکولو پروسـیسر (RISC) د 1990م لسیزې تر نیمایې پورې د کمپیوټر جوړونکوکمپنیو د پروســیســر د جوړولو لـه پـاره لـه پیچلي کمپیوټري

لارښـوونو سـیټ(CISC) تکنالوژي څخه ګټه اخیستله چې په همدې وخت کې ځینو د کمپیوټر جوړونکوکمپینو یو نوی نظر وړاندې کړو چې داسـې پروسـیسرونه باید جوړ شي تر څو وکولای شي په ډېر کم وخت کې د یو محدود اندازه لارښوونو د تر سره کولو وړتیا ولري چې همدې ټکنالوژی ته یې د RISC نوم ورکړ. او ځینې مشهور کمپیوټرونه یې له نوموړي پروسـیسـر سـره تولیـد کړل د بیلګې پـه توګـه : PowerPC او ARM او SPARC، د نوموړي ټکنالوژي لرونکو
پروسیسرونو یوه ګټه د سرعت زیاتوالې دی.

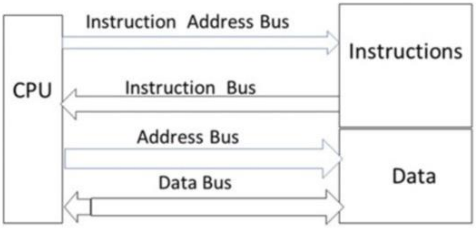
د RISC ټکنالوژي لرونکو پروسیسرونو ځینې ځانګړنې عبارت دي له:

۱. یو محدود تعداد لارښوونو (Instruction) ته اړتیا لري.
۲. ټولې لارښوونې (Instruction) یې د یو شان اندازې لرونکې دي.
۳. اکـثـره لارښــــوونــې(Instruction) پــه یــو دوران(Clock Cylce) کــې پروسس (Execute) کیږي.
۴. کنټرول واحـد (Control Unit) یې پروګرام شــوی نـه دی بلکې هـارډ وایرډ (Hardwired) شوی دی.
۵. د راجسټرونو زیاته اندازه لري.
۶. د څو محدودو آدرس حالتونو (Address Modes) لرونکي دي.

دا چې په یوه دوران کې د یوې لارښوی اجراکولو پروسیسر (RISC) ټکنالوژي لرونکي پروسیسرونه د کنټرول واحد له پاره هارډویر(Hardware) او پیچلي کمپیوټري لارښوونو سیټ(CISC) استفاده کوي نو له همدې امله په مرکزي پروسیس کوونکی واحد(CPU) کې کم ځای نیسې او د نوموړي پروسیسر طرح کوونکی (Designer) کولای شي له اضافه پاتې ځای څخه د راجسټرونو له پاره استفاده وکړی اما د پیچلي لارښوونې سیټ (CISC) ټکنالوژي لرونکو پروسیسرونو یوه لویه ګټه دا ده چې طرح کوونکي (Designer) کولای شي نوې لارښوونې په آسانۍ سره او بغیر له دې چې د پروسیسر جوړښت ته تغیر ورکړي اضافه کولای شي.



۴.٦ انځور د وانیومان(Von Neuman) جوړښت



۵.٦ انځور د هاروارډ (Harward) جوړښت

۴.٦ د پروسیس مرکزي واحد جوړښت (CPU Architecture)

په عمومي ډول سره دوه ډوله پروسیس مرکزي واحد(CPU) جوړښتونه شتون لري چې عبارت دي له وان نیومن (Von Neuman) او هاروارډ (Harvard) څخه او په لاندې ډول سره یې په لنډ ډول سره توپیر کولای شو.

۱. وان نیومن (Von Neuman) : نوموړی جوړښت له حافظې(Memory) څخه پروسیس مرکزي واحد(CPU) ته د ډیټا او لارښوونو د انتقال او لیږدونې له پاره د ډیټا وصلوونکې (Data Bus) څخه استفاده کوي لاندې شکل نوموړی جوړښت ښکاره کوي.

۲. هاروارډ (Harward) نوموړی جوړښت د ډیټا او لارښوونو له پاره بیلابیل

وصلوونکي (Buses) استفاده کوي ، د میموری څخه د لارښوونو د اخیستلو له پاره د لارښوونې ادرس وصلونکی (Instruction Address Bus) او لارښوونې وصلونکي (Instruction Bus) څخه استفاده کوي، په حافظه (Memory) کې د ډیټا د لیکلو او یا د ډیټا د اخیستلو له پاره ادرس وصلونکی (Address Bus) او ډیټا وصلونکی (Data Bus) استعمالیږي . 6.5 شکل په حافظې (Memory) کې د نوموړي جوړښت ښکارندويي کوي.

۵.٦ د اینټل کمپنی کوچني پروسیسر کورنۍ (Intel Microprocessor Family)

د اینټل (Intel) د پروسیسرونو د انټیګریټید سرکټ (IC) نمبر او نومونه په لاندې ډول سره یادولای شو:
8086, 8088,80, 286, 20,80, 386,80486 او Pentium I, Pentium II, Pentium III په دې وروستیو کې اینټل (Intel) او HP یو بل پروسیسر د Itanium په نوم سره تولید کړو او د دې کتاب په راتلونکو برخو کې به یې تر مطالعې لاندې ونیسو.

د کوچنی پروسیسر (MicroProcessor) ځانګړنې:

۱. *راجسټرونه (Registers):* د دې له پاره استفاده کېږي چې د پروسیسر په منځ کې معلومات وساتي او اندازه يې $8-bit, 16-bit, 32-bit, 64-bit$ ده
۲. *د راجسټرونو شمیر (Number of Register) :* د زیاتو راجسټرونو لرونکې پروسیس مرکزي واحد (CPU) کولای شي په خپل ځان کې ډېر زیات معلومات د پروسیس کولو له پاره وساتي.
۳. *د ډېټا وصلونکې اندازه (Data Bus Size):* دا څرګندوي چې په یو وخت کې د ډېټا څومره بیټونه په موازي ډول سره له میموري څخه ورکړېز او راکړېز

پـورټـونـو (Input/Output Port) تـه او یـا هـم لـه ورکـړیـز او راکـړیـز پـورټـونـو (Input/Output Port) څخه حافظې (Memory) ته انتقال شي.

۴. *د آدرس وصلونکې اندازه (Address Bus Size)*: د نوموړي اندازه دا معلوموي چې په یو وخت کې کوچني پروسیسر (Microprocessor) په حافظه (Memory) کې څومره ځای (Locations) ته لاس رسی پیداکوي.

۵. *د ساعت سرعت (Clock Speed)*: نوموړې ځانګړنه دا مشخصوي چې پروسیس مرکزي واحد (CPU) په یو وخت کې په څومره سرعت سره په ډېټا باندې عملیې تر سره کوي.

۶. *ریاضیکې کوپروسیسر (Math Coprocessor)*: د پروسیس مرکزي واحد (CPU) په منځ کې یو ځانګړی پروسیسر دی چې په مرسته یې د ریاضي پیچلي عملیې تر سره کېږي.

۷. *حقیقي حالت (Real Mode)*: د دې په مرسته د پخوانیو پروسیسرونو د ورژنونو (Versions) سره جوړښت او توافق منځ ته راځي.

۸. *ساتندویه حالت (Protected Mode)*: دا ځانګړنه چې له 80,286 پروسیسرو څخه وروسته د اینټل (Intel) په ټولو پروسیسرونو کې شتون لري چې په مرسته یې هر پروګرام ته د حافظې (Memory) یوه ځانګړې برخه ورکول کېږي چې نور پروګرامونه نه شي کولای د حافظې (Memory) دغه برخه استعمال کړي.

۹. *کیشي حافظه (Cache Memory)*: نوموړې حافظه (Memory) کوچنی او زیات سرعت لرونکې ده چې د پروسیس مرکزي واحد (CPU) او ریم (RAM) تر منځ موقعیت لري چې د موقت وخت د ډېټا د ساتلو له پاره استعمالېږي د کیشي حافظې (Cache Memory) د اندازې په زیاتیدو سره د پروګرام د ترسره کېدو سرعت زیاتیږي.

۱۰. *د MMX ټکنالوژي*: اینټل (Intel) نوموړې ټکنالوژي د دې له پاره استفاده کوي چې د ځینو کمپیوټري لوبو (Games) مطبوعاتي (Multimedia) او داسې نورو شیانو سرعت ورسره لوړ شي. نوموړی (SIMD) تخنیک څخه استفاده کوي چې کولای شي په یوه لارښوونه (Instruction) باندې ډېرې کړنې تر سره کړي.

Upward Compatibility د اینټل (Intel) جوړښت د Upward Compatibility لرونکی دی یعنې هغه پروګرام چې د ۱۶بیټه (bit) پروسیسر له پاره جوړ شوی دی کولای شو په ۳۲ بیټه (bit) پروسیسر باندې یې هم وچلوو.

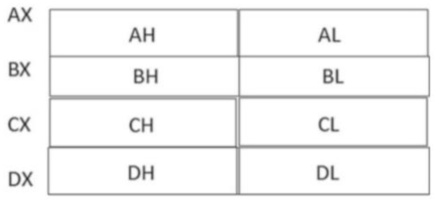
	80486dx	Pentium	Pentium Pro	Pentium Pro II	Pentium II
Register size	32 bits	32 bits	32 bits	32/64 bits	32/64 bits
Data bus size	32 bits	64 bits	64 bits	64 bits	64 bits
Address size	32 bits	32 bits	32 bits	32 bits	32 bits
Max memory	4 GB	4 GB	4 GB	4 GB	4 GB
Clock speed	25, 33 MHz	60, 166 MHz	150, 200 MHz	233, 340, 400 MHz	450, 500 MHz
Math processor	Built-in	Built-in	Built-in	Built-in	Built-in
L1 cache	8 KB, 16 KB	8 KB instruction 8 KB data	8 KB instruction 8 KB data	16 KB instruction 16 KB data	16 KB instruction 16 KB data
L2 cache	No	No	256 KB or 512 KB	512 KB	512 KB
MMX technology	No	No	Yes	Yes	Yes

۲.٦ جدول د اینټل کوچنې پروسیسر (Intel Microprocessor)

ځانګړتیاوې:

L1 کېشې هغه کیشې حافظه ده چې د کوچنې پروسیسر (Microprocessor) په داخل کې موقعیت لري

L2 کېشې د کوچنې پروسیسر (Microprocessor) برخه نه ده بلکې یوه ځانګړې اینټیګریټېډ سرکټ (IC) دی.

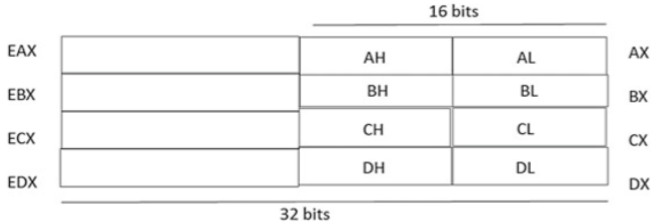


٦.٦ انځور د IA-16 راجسترونه

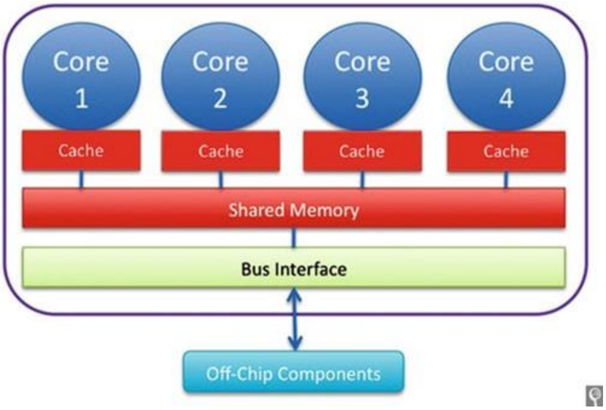
۷.٦ شکل د IA-32bit عمومی راجسترونه ښایي چې EAX,EBX,ECX او EDX راجسترونه 3-bit عمومي راجسترونه دي همدارنګه 6.6 شکل هم د IA-32bit راجسترونه ښکاره کوي نو هغه پروګرام چې د IA-16 له پاره لیکل شوی وی په IA-32 باندې هم چلیدلی شي.

٦.٦ د څو هستو لرونکو پروسیسر (Multi-Cores Processors)

د زیاتو هستو لرونکې پروسیسر داسې یو انټیګریټید سرکټ (IC) دی چې له یو څخه زیات مرکزي پروسیس کوونکي واحدونه (CPUs) پکې شتون لري چې کولای شي په یو وخت کې زیاتې لاربښوونې تر سره کړي او په ټوله کې د کمپیوټر په عمومي سرعت باندې مثبت تاثیر لري. د بیلګې په توګه یو Quade Core پروسیسر لرونکې کمپیوټر څلور بیلا بیل Cores لري چې هر یو یې د کړنو د تر سره کولو وړتیا لري.
لاندې شکل کې د کواډ کور (Quade Core) جوړښت روښانه کوي.
ځینی زیات هستی لرونکي پروسیسرونه د دوه هستو لرونکی پروسیسر (Dual Core CPU) لکه AMD Phenom II X2



۷.٦ انځور د IA-32-bit راجسترونه



۸.٦ انځور د زیاتو هستو لرونکې (Multi Core) پروسیر جوړښت

د درې هستو لرونکی پروسیسر (Tri Cores CPU) لکه AMD Phenom II X3

د څلور هستو لرونکی پروسیسر (Quade Cores CPU) لکه AMD Phenom II X4

د شپږو هستو لرونکی پروسیسر (Hexa-cores CPU) لکه AMD Phenom II X6

د اتو هستو لرونکی پروسیسر (Octa-Cores CPU) لکه Intel Core I7 5969X

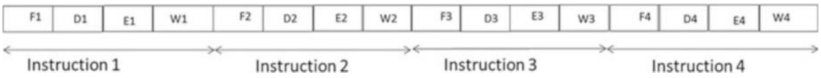
۷.٦ د پروسیس مرکزي واحد په واسطه د لارښوونو تر سره کول (Cpu Instruction Execution)

کله چې پروسیس مرکزي واحد (CPU) لارښوونې ترسره کوي نو لاندې مرحلې لري.

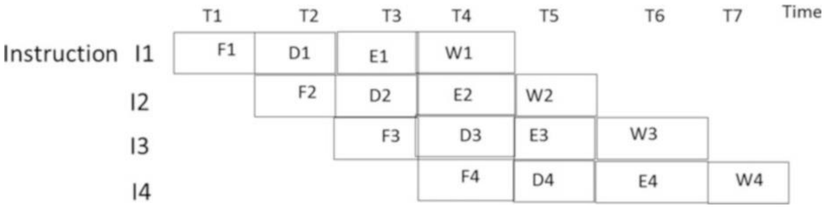
۱- د لارښوونو اخیستل (Fetch Instruction): له میموری څخه پروسیس مرکزي واحد(CPU) ته د لارښوونو انتقالول.

د لارښوونو ډیکوډ کول (Decode Instruction): په دې مرحله کې د لارښوونو ډول معلومېږي چې کوم ډول لارښوونې دي د بېلګې په ډول سره: AND, OR یا د NOT ,

او همدارنګه په دې مرحله کې که چېرته اړينه وي عامل (Operands) هم په راجسټرونو کې ساتل کيږي.



۹.٦ انځور د پايپ لاين څخه پرته د لارښوونو تر سره کول



۱۰.٦ انځور: له پايپ لاين سره د لارښوونو تر سره کول

۲- د لارښوونو تر سره کول (Execution) د بيلګې په توګه د جمعې لارښوونې.

۳- په ريجيسټرونو يا هم زيرمه کې د پايلو ساتنه (Storing).

پايپ لاينينګ (Pipelining): کولای شـــي چې د پروســـيس مرکزي واحد (CPU) سرعت او ځرنګوالي ښه کړي لکه څرنګه چې تاسو په ۹.٦ شکل کې ګورئ بغير له پايپ لاينينګ څخه د پروسـيس مرکزي واحد (CPU) د هرې لارښـوونې (Instruction) له پاره T ثانيې وخت نيسي چې په ټوله کې $16\,T$ وخت نيسي او په ۱۰.٦ کې لکه څرنګه چې تاسـو ګورئ له پايپ لاينينګ سـره د پروسـيس مرکزي واحد (CPU) لارښـوونې تر سره کوي او د پروسيس مرکزي واحد (CPU) د يوې لارښوونې د تر سره کولو په جريان کې راتلونکې لارښـوونه تعقيبوي کوې. دا چې په يو وخت کې د پروسـيس مرکزي واحد (CPU) يوه لارښـوونه باندې کړنې تر سره کوي او بله لارښوونه له حافظې (Memory) څخه تر لاسه کوي د دې له پاره اړينه ده تر څو دوه کېښې حافظه(Cache Memory) هم ولرو چې يو ته يې د معلوماتو حافظه (Data Memory) او بل ته يې لارښـــوونې

حافظه (Instruction Memory) وايي او د پروسیس مرکزي واحد (CPU) نوموړي جوړښت ته هاروارډ جوړښت وايي.

په مستقیم ډول سره میموري ته لاسرسی (Direct Memory Access): د نوموړي تخنیک څخه په استفادې میموري کولای شي په مستقیم ډول سره راکړیزو او ورکړیزو آلو (Input/output Devices) سره د ډېټا راکړه ورکړه تر سره کړي مطلب دا چې په عادي حالاتو کې له میموري څخه د ډېټا د هر بلاک د تر لاسه کولو له پاره باید چې د پروسیس مرکزي واحد CPU نوموړې کړنه تر سره کړي او راکړیزو او ورکړیزو آلو (Input/output Devices) ته يې ولېږدوي چې دا زیات وخت نیسي، مګر په مستقیم ډول سره میموری ته لاسرسي (DMA) په مرسته کولای شو چې په مستقیم ډول سره د له میموري څخه ډېټا ورکړیزو او راکړیزو آلو (Input/output) ته ولېږدوو.

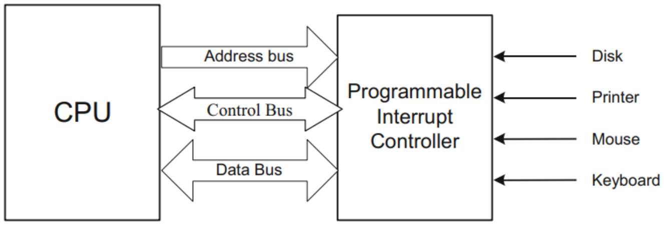
په مستقیم ډول سره میموری ته لاسرسي (DMA) په لاندې ډول سره کار تر سره کوي:

- په مستقیم ډول سره میموری ته لاسرسي (DMA) د پروسیس مرکزي واحد (CPU) ته یو غوښتونکې (Request) سیګنل لېږدوي.
- دپروسیس مرکزي واحد (CPU) نوموړې غوښتنې ته د یو اجازی(Grant) ریکویسټ په شکل سره ځواب ورکوي او د وصولونکې (Bus) د استفادې اجازه ورته ورکوي.
- په مستقیم ډول سره میموري ته لاسرسی (DMA) د وصلونکې (Bus) اداره په خپله غاړه اخلي او وروسته له دې میموري کولای شي په مستقیم ډول سره راکړیز او ورکړیزو آلو(Input/output Devices) ته ډېټا ولېږدوي او یا هم ډېټآ ورڅخه تر لاسه کړي.
- په مستقیم ډول سره حافظې (Memory) ته لاسرسي (DMA) دا هم کولای شي چې له یو خارجې آلې(External Device) څخه یو لوی فایل حافظې (Memory) ته بغیر د پروسیس مرکزي واحد (CPU) له اجازې څخه راوړې او غوښتل شوې کړنه ورباندې تر سره کړي.

د يـادونې وړ ده چې پـه مســتقيم ډول ســره حـافظې (Memory) ته لاســرسـي (DMA) نسـبت د پروسيس مرکزي واحد (CPU) ته ډېر د لوړ سرعت لرونکی او موثر تخنيک دی.

پـروګـرام مـنـونـکـې راکـړيـز ورکـړيـز آلو مزاحـمـت (Programmable I/O Interrupt)

دا چې له کمپيوټر سره بيلا بيل راکړيزی او ورکړيزې آلې (Input/output Devices) نښلول شوي دي او نومونړي هره يوه آله خپل حالت د پروسيس له مرکزي واحد (CPU) سره شريکول غواړي چې دغه کار د پروګرام منونکې راکړيزو او ورکړيزو آلو مزاحمت (Programmable I/O Interrupt)په مرسته ترسره کيږي تر څو د هرې آلې تازه حالت مرکزي پروسيس کوونکی (CPU) ته ورسوي او وروسته له هغې د پروسيس مرکزي واحد(CPU) په مرسته د يو مسيج په ډول سره له استفاده کوونکي شخص سره شريک کيږي د بيلګې په توګه کله چې په پرينټر (Printer)کې کومه ستونزه موجوده وي، پرينټر (Printer) نوموړې ستونزه په مستقيم ډول سره د پروسيس له مرکزي واحد (CPU) سره نه شي شريکولای د دې له پاره لومړی پروګرام منونکې راکړيز او ورکړيز مزاحمت (Programmable I/O Interrupt) ته خبر ورکوي او له هغې وروسته پروګرام منونکې راکړيز ورکړيز مزاحمت (Programmable I/O Interrupt) د پروسيس مرکزي واحد(CPU) ته خبر ورکوي.



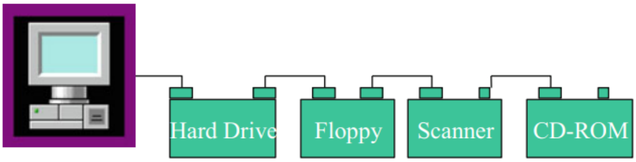
۱۱.٦ انځور Programmable I/O Interrupt

۸.٦ د ډېسک اداره کوونکی (Disk Controller)

نوموړې آله د ډېسک یوه برخه چې هېډ (Head) نومیږي محرکه کوي تر څو په مرسته یې ډېټا ولوستل شي او یا هم ډېټا ولیږدول شي. نن سبا دوه ډوله ډېسک اداره کوونکی (Disk Controller) استفاده کیږي چې عبارت دي له:

- **الکترونیکې انټیګریټډ ډیسک (Integrated Disk Electronics) IDE:** د IDE Disk Drive د ISA بس سره د یو کېبل په واسطه نښلول شوي دي. دا چې پوهیږو چې د IDE ډیسک اداره کوونکی کولای شي تر دوه پورې داسې هارډیسکونه قبول کړي چې اعظمي ظرفیت یې 528 MB وي نو له همدې امله په ۱۹۹۴ م کال کې د هارډیسک تولید کوونکو کمپنیو د EIDE یعنې غځیدلې IDE تکنالوژي معرفي کړه چې د لومړي ځل په په IBM کمپیوټرونو کې استفاده شوه او کولای یې شو تر څلورو پورې آلې قبولې کړي لکه : هارډیسکونه، ټېپ ډرایونه، سي ډي روم او لوی هارډیسکونه نوموړې ټکنالوژي لرونکی ډیسک اداره کوونکي کولای شي دوه هارډیسکونه چې د هر یو ظرفیت تر 250 GBs پورې وي قبلولای شي.

د کوچني کمپیوټر سیستم مخینه (SCSI): نوموړې تکنالوژي د لومړي ځل له پاره د ANSI په واسطه معرفي شوه چې د راکړیزو او ورکړیزو آلو (I/O Devices) د وصل کېدلو له پاره استفاده ورڅخه کیږي لکه د سکېنر (Scanner) هارډیسکونه او سي ډي رومونه او داسې نور. نوموړی د ټولو مایکروکمپیوټرونو له پاره سټنډرډ انټرفېس دی چې د مکنټاش، RISC Workstation او IBM کمپیوټرونه یې د استفادې ښې بیلګې دي.



۱۲.٦ انځور SCSI موډل

	Bandwidth	Data rate MB/s
SCSI-1	8 bits	5
SCSI-2	16 bits	10–20
Ultra SCSI	8 bits	20
SCSI-3	16 bits	40

MB/*s* millions bytes per second

۳.٦ جدول د SCSI کنټرول د مختلفو ډولونو خاصيتونه

۱۳.٦ انځور: د SCSI ډولونه

- **SATA (Serial Advanced Technology Attachment)**: نـومـوړي ټکنالوژي د کمپيوټر سره مسلسلو آلو (Devices) د نښلولو له پاره استفاده کېږي.

۹.٦ کوچني کمپيوټر وصولونکی (Microcomputer Bus)

د کوچني کمپيوټرونو له پاره نن سبا په مارکيټ کې بيلا بيل وصولونکي (Buses) اسـتـفـاده کـېږي چـې عبـارت دي لـه: ISA, MCA, EISA, VESA PCI, FireWire, USB او PCI څخه. چې موږ په دې کتاب کې يوازې دوه ډولونه USB

او PC Express په پوره ډول سره تر څیړنې لاندې نیسو.

۱.۹.٦ د صنعت معیاری جوړښت (Isa Bus) وصلونکی: د د صنعت معیاری جوړښت (ISA) وصلونکې (Bus) د لومړي ځل له پاره د(IBM) کمپنی په واسطه د (IBM) د ۸۰۸۸ پروسیسر لرونکو کمپیوټرونو له پاره استفاده شو. نوموړی بس یو ٦- بیټ ډېټا وصولونکی لري او ۲۰ لاینونه د آدرس له پاره لري او د کلاک سرعت یي ۸ مېګا هرتز (8MHz) دی.

۲.۹.٦ د کوچنې چېنل جوړښت وصلونکې (MCA): نوموړی بس په ۱۹۸۷م کال کې د IBM کمپنی په واسطه د PS/2 کمپیوټر له پاره استفاده شو چې یو ۳۲- بېټ وصولونکی(Bus) دی او کولای شي په یو وخت کې ۴ بایټه ډېټا د ۱۰ مېګا هرتز په کلاک سرعت سره انتقال کړي. د یادونې وړ ده چې نوموړی وصلونکی د ۱٦-بېټ ډېټا وصلونکی او ۳۲-بېټ آدرس لاینونو په لرلو سره هم کړنه تر سره کولای شي. دا چې نوموړي بس قېمت ډېر زیات وو له همدې امله له IBM کمپنی څخه پرته نورو تولید کوونکو کمپنیو یو بل ډول وصلونکې(Bus) چې د (EISA)په نوم سره یادېږي منځ ته راوړ.

۳.۹.٦ د (EISA) وصلونکې (Bus): نوموړی وصلونکې (Bus) اصلا یو 32-bit وصلونکې(Bus) دی چې کولای شي چې د ۸-بېټ او ۱٦-بېټ د ډېټا د لېږدوي د ساعت سرعت یې ۸ مېګا هرتز (8MHz) دی او همدارنګه د ۳۲-بېټ آدرس لاینونو درلودونکی دی.

۴.۹.٦ د ویزا (VESA) وصلونکې (Bus): د ویسا (Video Electronics Standard Association) وصلونکی چې همدارنګه د (Video Local Bus) (VL-BUS)په نوم سره هم یادېږي د کمپیوټر او د کمپیوټر د ایکسپینژن(Expansion) تر منځ د یو حایل (Interface) په ډول سره کار کوي دا چې څومره اپلیکېشن د ویډیو پر اساس مغلق کېږي په هماغه اندازه یې د

نوموړي وصـــلونکي(Bus) له پاره اړتیا زیاتېږي او نوموړی وصـــلونکې (Bus) کولای شي چې د سټیشنونو (Stations) تر منځ په تیزی سره ډېټا انتقال کړي. او تر MBPS 132 پورې د ډېټا د انتقال وړتیا لري.

۵.۹.٦ د پی سی ایي (PCI) وصلونکې (Bus): نوموړی وصلونکی (Bus) د لومړي ځل لـه پاره د اینټل (Intel) کمپنی په واســطه منځ ته راغلی چې له ۳۲-۶۴ بېټ پورې وصـلونکي(Buses) لري او د ۳۳-٦٦ مېګا هرتز سـاعت سـرعت لري. د یـادونې وړ ده چې نوموړی وصــلونکی(Bus) د بیلا بیلو آلو لکـه : ,Switch Router, Hub سره د وصل کېدلو له پاره ډېر موثر دی.

Bus type	ISA	EISA	MCA	VESA	PCI	PCI-٦٤
Speed (MHz)	٨	٨,٣	١٠	٣٣	٣٣	٦٤
Data bus bandwidth (bits)	١٦	٣٢	٣٢	٣٢	٣٢	٦٦
Max. data rate (MB/s)[a]	٨	٣٢	٤٠	١٣٢	١٣٢	٥٠٨
Plug and play capable	No	No	Yes	Yes	Yes	Yes

۴.٦ جدول د بیلا بیلو بسونو ځانګړنې

۱۴.٦ انځور PCI کارت

د یادونې وړ ده چې (PCI) وصــلونکی (Bus) د دې له پاره منځ ته راغی چې د کمپیوټري سیستم (Bandwidth) زیات او ځنډتیا (Latency) یې کمه شي. د نوموړي وصلوونکي (Bus) اوسني ورژنونه تر ۱۰۵٦ میګابایټ پورې ډېټا ریټ لري چې تر ۴۲۲۴ میګـابـایـټ پورې د غځیـدلو وړتیا هم لري. د (PCI) بس ۱٦ پورې پورټونه(ports) لري یعنې کولای شـي په مادربورډ(Motherboard) کې تر ۱٦ پورې آلې له خپل ځان سره

ونښلوي. ډيری د (ATM) ماشينونه نوموړي وصلونکی(Bus) له حايل (Interface) څخه استفاده کوي.

6.4 جدول د وصلونکې (Bus) د بيلا بيلو جوړښتونو لکه ,ISA, EISA, MCA VESA او PCI ځانګړتياوې ښکاره کوي او 6.14 شکل د (PCI) وصلونکی ښکاره کوي.

۶.۹.۶ يو ايس بی(Universal Serial Bus (USB: نوموړی وصلونکی(Bus) يو کمپيوټري مسلسل وصلونکی (Bus) دی چې کولای شي اکثره آلې لکه سکنر، پرينټر، موډيم، کيبورډ، موس او داسې نور بغير له کومی اضافي کړنې تر سره کولو له کمپيوټر سره ونښلوي د دې څخه مطلب دا دی چې د (USB) وصلونکې (Bus) د دې وړتيا لري چې له ځان سره نښلوونکې آلې تشخيص کړي او د هغه د نښلولو او بيرته ويستلو حالت کمپيوټر ته ورکړي. هغه شخصي کمپيوټرونه چې د (USB) پورټونه(Ports) لري کولای شي د بيلا بيلو آلو سره ونښلي. او همدارنګه (USB) کولای شي دا معلومه کړي چې کومه آله ورسره وصل شوې ده. د يادونې وړه چې په يو وخت کې کولای شو له يو (USB) سره تر ۱۲۷ پورې آلې وصل شي. د 1.1 ورژن (USB) په ۱۹۹۸م کال کې منځ ته راغله چې د ډېټا د انتقال او لېږدونې سرعت يې

12 Mbps په لوړه کچه او 1.5 Mbps په ټيټه کچه باندې وو چې دا ښکته سرعت يې د ځينو آلو لکه موس، کيبورډ او Joystick له پاره استفاده کېدلو

USB 2 .0په ۱۹۹۹م کال کې منځ ته راغلو چې سرعت يې تر 480 Mbps پورې رسيږي د يادونې وړ ده چې د USB له پاره اعظمي د کيبل اوږدوالې تر ۵ ميترو پورې رسيږي.

۷.۹.۶ د يو ايس بی جوړښت (Usb Architecture): د ۱۵.۶ شکل موږ ته د USB جوړښت څرګندوي چې د USB جوړښت په منطقي لحاظه د Tree

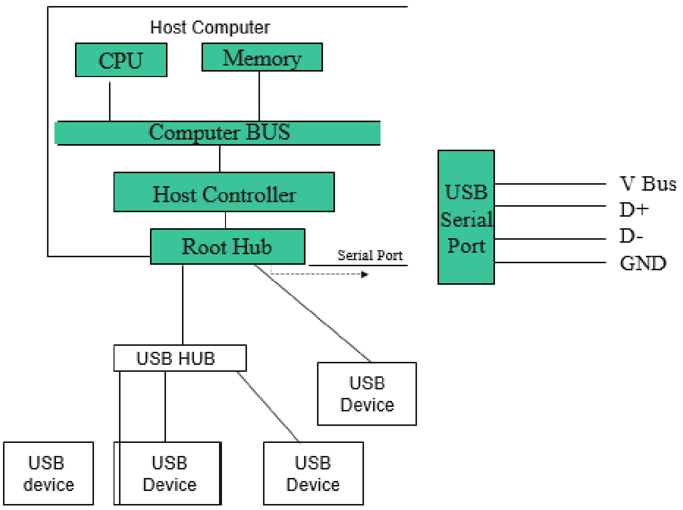
ټوپولوژي او لـه فزیکي لحـاظـه د Star ټوپولوژي لري چې لـه همـدې املـه هر USB آله له Root هب ســره نښــلي چې په هر USB ســیســتم کې یو کوربه (Host) کنترولر موجود وي. په یو USB سیستم کې لاندې برخې شتون لري: د USB Host اداره کوونکی، د USB اصــلي هــب (USB Root Hub)، USB Hub، د USB کېبـل، USB آلـه، د کلاینټ ســافټویر او همدارنگه د هاسټ اداره کوونکی سافټویر.

کوربه اداره کوونکی (Host Controller)

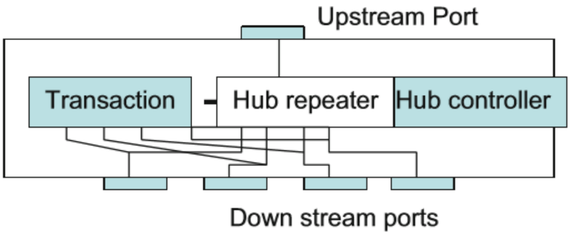
کوربه اداره کوونکی (Host Controller) د ټولو معلوماتو لېږد پیل کوي او روټ هـب (Root Hub) د کـوربـه کـنـترولونکی (Host Controller) او د آلو (Devices) تـرمـنځ اړیکـه چمتو کوي روټ هـب د کوربـه اداره کوونکی (Host Controller) لـه خوا رامینځتـه شــوی لېږد تر لاســه کوي او د یو اس بی (USB) الو (Devices) تـه یـې لېږدوی کوربـه اداره کوونکی (Host Controller) پولینـگ (Polling) ددې له پاره استعمالوي ترڅو معلوم کړي چې کومه نوې آله (Devices) له وصلونکي (Bus) سره وصل ده او کومه رابطه له وصلونکي (Bus) سره پرې شوي ده.

د USB Host اداره کوونکی لاندې کړنې تر سره کوي.

۱. د Host اداره کوونکي یوه آلـه د عمـل کولو لـه پـاره آمـاده کوي (Device Configuration).
۲. د Packet منځ ته راوړل.
۳. Serialize او Deserializer
۴. له آلې Device او Host څخه د تقاضا پروسیس کول
۵. د USB پروتوکول اداره کول



۱۵.٦ انځور د USB جوړښت



۱٦.٦ انځور د HUB جوړښت

USB version	Release data	Data rate	Data rate designation
USB 1.0	1996	1.5 Mbps	Low speed
USB 1.1	1998	12 Mbps	Full speed
USB 2.0	2000	480 Mbps	High speed
USB 3.0	2008	5 Gbps	Super speed
USB 3.1	2013	10 Gbps	Super speed þ

۵.٦ جدول د یو اس بی (USB) ورژنونه او ځانگړنې

۶. د Host او USB آلو تر منځ د ډېټا د جریان کنترولول
۷. آلو ته د آدرسونو ورکول
۸. د Client سافټویرونو په کار راوستل
۹. له USB پورټونو څخه د بېټ حالت (Status Bit) راټولول.

روټ هب (Root Hub) د دی وظیفه داده چې آلو ته برېښنا لېږدوي پورټونه (Ports) فعال او غیر فعالوي او د پورټونو حالت host اداره کوونکي ته لېږدوي. Root Hub د Host د اداره کوونکي او د USB تر منځ اړیکه جوړوي

عمومي وصلونکی کېبل (USB Cabel): د 6.15 شکل د یو USB نني جوړښت ښکاره کوي چې د ۴ سیمونو درلودونکی دی چې د V وصلونکی یې د برېښنا د لېږدولو له پاره استفاده کېږي او +D او -D د سیګنل د لېږدولو له پاره استفاده کېږي.
USB Devices د USB آلې په بیلا بیلو طبقو تقسیم شوې دي لکه HUB, Printer, Mass Storage او هره آله ځینې معلومات لري لکه: د آلې نوم، د جوړونکي آی ډي، کلاس او ډېټا رېټ. د هاسټ اداره کوونکی نوموړي معلومات د دې له پاره استفاده کوي چې له ډېسک څخه د یوې آلې ډرایور راواخلي.

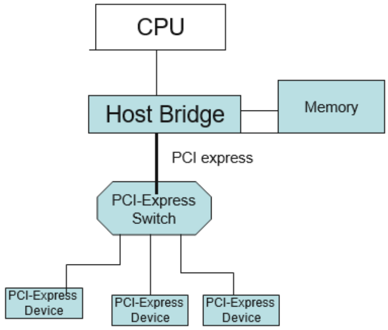
USB آلې کولای شي بیلا بیلې کړنې ولري لکه په یو سپیکر کې د غږ اداره او داسې نور د یادونې وړ ده چې په USB کې هره کړنه د یو Endpoint آدرس په مرسته ځانګړې کېږي.

۸.۹.۶ د وسایلو ترمنځ وصلونکي PCI Express Bus: نوموړی وصلونکی د لومړي ځل له پاره په ۱۹۹۰م کال کې له ۳۳ میګا هرتز فریکوینسي سره معرفي شو او د وخت په تیریدلو سره یي سرعت ٦٦ میګا هرتز فریکونسي زیات شو. د نیټورک په نوو منځ ته راغلو تغیراتو سره او Gigabit Ethernet او داسې راکړه یز او ورکړه یزو آلو (Input/output Devices) په منځ ته راتلو سره چې د زیاتو ځانګړنو لرونکی وې اړتیا وه تر څو یو نوی او د لوړو ځانګړنو لرونکی

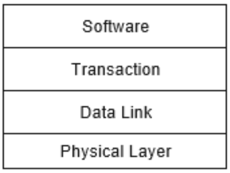
بس منځ ته راشی چې له همدې امله په ۲۰۰۲م کال کې د Special Interest ګروپ په واسطه PCI Express تایید شو او په ۲۰۰۴ م کال کې یې Chipset کې ځای پر ځای کړ چې د لاندې ځانګړنو درلودونکی دی:

- د دوه آلو تر منځ نقطه په نقطه (Point to Point) اړیکه جوړوي.
- PCI Express یو مسلسل وصلونکی دی.
- PCI Express د Packet او Layer جوړښت څخه استفاده کوي.
- د سافټویرونو څخه په استفادې له PCI وصلونکې سره هماهنګي جوړوي.
- د تیروتنې د پیدا کولو وړتیا لري.
- په بیلا بیلو وختونو کې د معلوماتو لیږدونه تر سره کولای شي.
- انتخاب شوی باندویټ (Bandwidth)

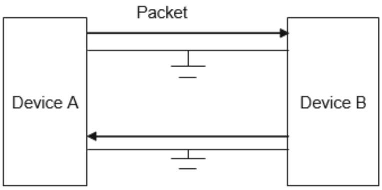
۶. ۹.۹ د وسایلو ترمنځ د وصلونکي جوړښت (PCI Express Architecture): د ۱۸.٦ شکل د نوموړي وصلونکي جوړښت څرګندوي په داسې ډول سره چې Host Bridge د حافظې او PCI Express سویچ (switch) تر منځ اړیکه ټینګوي او یوې مخینې په شکل کړنه تر سره کوي. د PCI Express Switch ددی له پاره استعمالیږي چې د وسایلو تر منځ وصلونکي (PCI) وټو او ننوتو د ځایونو (port) تعداد زیات کړي.



۱۷.٦ انځور د PCI Express وصلونکي جوړښت



۱۸.٦ انځور د PCI Express د بس پروتوکول



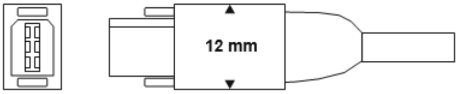
۱۹.٦ انځور د PCI Express اړیکې

۱۰.۹.٦ **د PCI Express پروتوکول**: لانـدې شــکـل کې د نوموړي بس عمومي پروتوکول کتلای شي چې په څلورو برخو باندې تقسیم شوی دی. د پوستغالۍ طبقه (software layer) د پورته پروتوکول لومړی برخه پوستغالی (software) د دې له پاره استفاده کېږي چې د PCI Express او هغه آلو تر منځ چې له نوموړي بس سره وصلوي هماهنگي رامنځ ته کړي.

فزیکي طبقه (Physical Layer) دا ښکاره کوي چې له PCI Express سره دوه آلې د PCI Express Link (Lane) په مرسته وصل شوي دي هر lane څلور وایرونه لري او د PCI Express هر یو Lane دوه Simplex اړیکې لري چې یو یې د Packet د لېږدولو او دوهم یې د Packet د تر لاسه کولو له پاره استفاده کېږي. د یادونې وړ ده چې د PCI Express تر 2.5GPS پورې د ډېټا د لېږدولو وړتیا لري. د PCI Express د لینک د X1, X2, X4, X16 او X32 په شکل سره تخته په تخته برابرولی (Configure) او جوړولای شو چې له X1 څخه هدف دا ده چې 1 Lane د څلورو وایرونو لرونکی او X4 څخه هدف دا ده چې 4 Lanes او 16 وایرونو لرونکی دی.

۱۰.۶ فايروال (Firewire)

فاير واير يا IEEE 1394 يو لوړ سرعت لرونکی مسلسل وصلونکی دی چې د ډيجيټل آلو لکه ډيجيټل ويډيو او کامکوډر (Camcorder) د وصل کولو له پاره استفاده کېږي، نوموړی وصلونکې د دې وړتيا لري چې په 100Mbps, 200Mbps, 400Mbps سرعت سره ډيټا ولېږدوي. IEEE 1394 کېبل د ۶ مسي لاينونو درلودونکی دی چې دوه وايرونه يې د برق انتقال او ۴ وايرونه يې د سيګنل انتقال تر سره کوي ځينی د فاير واير وصلونکی بيا د ۴ لاينونو لرونکی دی چې ټول د ډېټا د انتقال له پاره استفاده کېږي. لاندې شکل د فاير وال مذکر او مونث ښکاره کوي.



۲۰.۶ انځور مذکر او مونث نښلوونکی

Pin	Signal name	Description
1	Power	Unregulated DC; 17–24 V no load
2	Ground	Ground return for power and inner cable shield
3	TPB—	Twisted-pair B, differential signals
4	TPBþ	Twisted-pair B, differential signals
5	TPA—	Twisted-pair A, differential signals
6	TPAþ	Twisted-pair A, differential signals

۶.۶ جدول IEEE 1394 pins

۱.۱۰.۶ ايچ ډی ايم ايي HDMI (High Definition Multimedia Interface):

د نا فشار شويو ويډيو او فشار شويو غږيزو معلوماتو د انتقال له پاره استعماليږي. د HDMI ځينې کارونکي عبارت دي له د کمپيوټر مانيټور، ډيجيټل ټيلويزيون، ويډيو پروجيکټور. لاندې شکل د HDMI بيلا بيل ډولونه ښکاره کوي.

۲۱.٦ انځور د HDMI نښلوونکې

د کمپیوټر مورني تخته (Motherboard) هغه جوړ شوی سرکټ لرونکي تخته(board Printed Circuit) ده چې د کمپیوټر زیاتی برخې پکې ځای پرځای شوی دي لکه ,CPU ,RAM ,ROM USB Expension slot and چې ۲۲.٦ شکل د کمپیوټر مورنۍ تخته (MotherBoard) شکل ښایي.

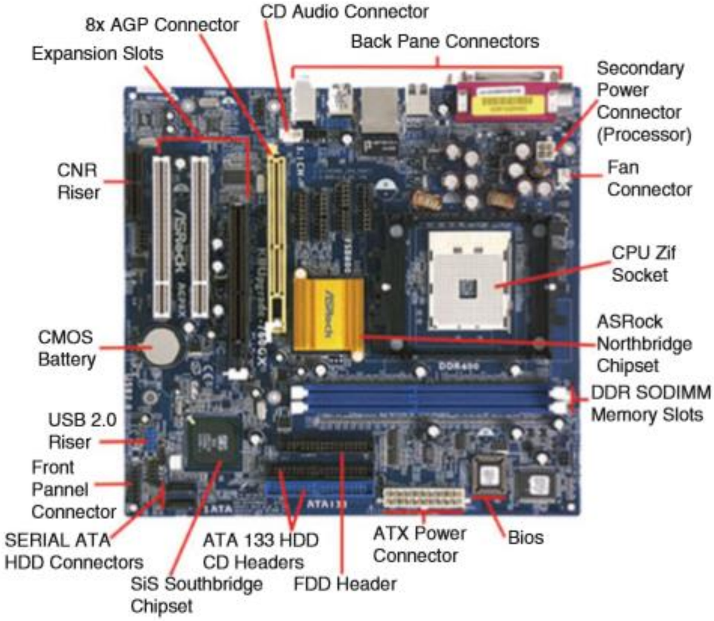
۱۱.٦ خلاصه

- د پروسیس مرکزي واحد (CPU) ، حافظه، موازی راکړیز او ورکړیز، پرله پسې راکړیز او ورکړیز د پروګرام کولو مزاحمت او ډی ام ای (DMA) د کوچني کمپیوټر له اجزاوو څخه دي.
- د لارښوونو اجرا کول د پروسیس مرکزي واحد (CPU) له دندو څخه دي.
- حسابي او منطقي لاجیک برخه (ALU)، د کنټرول برخه، او راجسترونه د پروسیس مرکزي واحد (CPU) له اجزاوو څخه دي.

اکثره کمپیوترونه درې قسمه حافظه استعمالوي، کېشې حافظه (SRAM)، اصلي حافظه (DRAM or SDRAM)، او ثانوي حافظه (Hard Disk, Tape Drive, And Floppy Disk).

- د نیمه هادی حافظې اقسام عبارت دي له DRAM, SDRAM, EDORAM, DDR, SDRAM, RDRAM, ROM, and EPROM.
- SRAM په کېشې حافظه کې استعمالیږي، DRAM او SDRAM یه اصلي حافظه کې استعمالیږي.
- SATA, SCSI-1, SCSI-2 او SCSI-3 د کمپیوټر کنټرولونکې وسایل دي.

۲۱.٦ د اچ ډي ام ای (HDMI) وصلوونکي



۲۲.٦ انځور د مادر بورډ تصویر

ISA وصـــلوونکـی، MCA او PCI دا ټول مـایکرو کمپیوټر وصــلوونکی دی او Firewire یو یو تیز او د ډیری ډیتـا د وړلو قـابلیت لرونکی دی چی تر 400 Mbps وړتیا لری.

- USB یو سریال وصلوونکی(Serial Bus).
- PCI یو سریال صلوونکی ښکاره کوی.
- اووم څپرکی نیمـه هـادي میموري،هـاردیســک (Hard disk) -Solid-State disk، د کیشی کړنلارې میتود.

(Cache Mapping Mehtod)، اوخیـالي حـافظـه Virtual Memory تر بحث لاندې نیسي.

څو ځوابه پوښتنې

۱. ______ د ریاضیکې عملیو د سرته رسولو دنده اجرا کوي

a) وصلونکی(Bus)
b) مسلسل پورټ (Serial Port)
c) ALU
d) کنټرول واحد

۲. کله چې تاسو د پروسیس مرکزي واحد (CPU) او کوچني پروسیسر وظایف مقایسه کوي

a) دواړه یو شان کار کوي
b) دواړه یو شان کار نه کوي
c) د پروسیس مرکزي واحد (CPU) له کوچني پروسیسر څخه تیز دی
d) کوچنی پروسیسر د پروسیس له مرکزي واحد (CPU) څخه تیز دی

۳. د RISC پروسیسر ______ استعمالوي

a) Complex Instruction Sets
b) Reduced Instruction Sets
c) a and b
- None Of The Above

۴. د CISC پروسیسر د کنټرول برخه ______ده

a) سختغالی (Hardware)
b) میکرو کوډ (Microcode)
c) a او b
d) پورتني یو هم نه دی

۵. د ______ استعمال نه غیر هم حافظې ته مستقیم لاسرسي تاسو له حافظې څخه ورکړیزو او راکړیزو الو ته د معلوماتو د یوه بلاک د انتقال اجازه در کوي

a) د پروسیس مرکزي واحد (CPU)

b) د معلوماتو د تنظیمونکې (Data Bus)

c) کنترول بس (Control Bus)

d) حافظې ته مستقیم لاسرسي کنترولونکی (DMA Controller)

۶. کوم یو د لاندېنیو نه 32 بیټه وصلونکی دی ؟

a) ISA

b) PCI او EISA

c) EISA او ISA

d) MCA او ISA

۷. کوم یو د لاندې عامل سیستمونه د سختغالو خپل منځ کې توافق (Plug And Play) حمایه کوي؟

a) Window NT And Window 95

b) Window 98 And Window NT

c) Window XP And Window 2000

d) DOS And Windows NT

۱. د کوچنی کمپیوټر اجزاوي لست کړئ.

۲. د پروسیس مرکزي واحد (CPU) وظایف تشریح کړئ .

۳. د کنټرولونکې برخې (ALU) دنده څه ده ؟

۴. د پروسیس مرکزي واحد (CPU) وظایف لست کړئ؟

۵. نیم لغت څومره بیټه کیږي ؟

۶. پوره لغت څومره بیټه کیږي؟

۷. د پروسیس مرکزي واحد (CPU) او کوچنی پروسیسر تر منځ فرق واضح کړئ؟

۸. د معلوماتو د وصلیدونکي او ادرسونو د وصلیدونکي دندی تشریح کړئ؟

۹. حافظې ته مستقیم لاسرسي (DMA) دندی تشریح کړئ؟

۱۰. د موازي سوریو (Parallel Port) کاریالونه کوم کوم دي ؟

۱۱. د پرله پسې سوریو (Serial Port) کاریالونه کوم کوم دي؟

۱۲. د پوستغالو د اجرا په وخت کې مزاحمت (Interrupt) څه ته وايي ؟

۱۳. د پروسیس مرکزي واحد (CPU) له پاره زیاته حافظه د 16 ادرسونو لینونو او 8 معلوماتو لینونو سره کومه ده؟

۱۴. د CISC پروسیسر خصوصیات لست کړئ؟

۱۵. د RISC پروسیسر خصوصیات لست کړئ؟

۱۶. د نیومن (Neumann) او هارورډ (Harvard) له نظریو د جوړښت تر منځ فرق واضح کړئ؟

۱۷. د یو ګونی کور(Single Core) پروسیس مقابل کې د ملتي کور (Multicore) پروسیسر فایدی کومې دي؟

۱۸. د پروسیس مرکزي واحد (CPU) د لارښوونې د اجرا کولو ګامونه لست کړئ؟

۱۹. پیچ لارښوونې (Fetch Instruction) تشریک کړئ؟

۲۰. د پروسیس مرکزي واحد (CPU) د پنځو لارښونو د اجرا کولو له پاره د پیپ لینو په استعمال سره څومره وخت نیسي که چیري هر ګام د پیپ لینونو 20 دقیقې وخت ونیسي؟

۲۱. د بي پیپ لینه (Non-Pipeline) پروسیسر په استعمال سره د 21 مشکل د اجرا وخت محاسبه کړئ؟

۲۲. د 64 بیټه پروسیسر خصوصیات کوم کوم دي؟

۲۳. د ډیسک کنترولر اقسام لست کړئ؟

۲۴. د کمپیوټر وصلونکې لست کړئ؟

۲۵. دوه پرله پسې وصلونکې لست کړئ؟

۲۶. یو کمپیوټر زیات نه زیات څومره پرله پسې عمومی وصلونکې (USB) سوري لري؟

۲۷. د پرله پسې عمومی وصلونکې (USB) سوري وښایست؟

۲۸. د PCIe lane ډیاګرام وښایست؟

۲۹. د Firewire کاریال کوم یو دی؟

۳۰. HDMI کاریال کوم یو دي؟

اووم څپرکی

حافظه

(Memory)

موخې: ددی څپرکي پای ته رسیدو سره به تاسو لاندې کارونه وکولی شئ:

- د نیمه هادی حافظې (Semiconductor Memory) او د هغی د اقسامو تشریح کول.
- په مرستندویه زېرمه (Hard Disk) کې د سیکټر (Sector) او ټریک (Track) تشریح کول.
- د مرستندویه زېرمه (Hard Disk) د ظرفیت اندازه کول.
- د حافظې (Memory) د تسلسل (Hierarchy) زده کول.
- د Cache Miss, Cache Hit او Cache Hit Ratio زده کول.
- په کمپیوټر کې د حافظې (Memory) د اقسامو تشریح کول.
- په کېشی حافظه (Cache Memory) کې د مرستندویه حافظې څخه د محتویاتو راوړل.
- کېشی حافظې (Mapping) د کړنلارو سره اشنایي.
- د خیالې ادرس (Virtual Address) اړول فزیکې ادرس (Physical Address) ته.

- د صفحو جدول (Page Table) د وظایفو سره اشنايي.
- د خیالي ادرس (Virtual Address) څخه د فزيکي ادرس (Physical Address) لاسته راوړل.

۱.۷ سریزه

په يو کمپیوټر کې حافظه (Memory) ډیټا او هدايات (Instructions) زېرمه کوي، او د کمپیوټر د يوې ډېرې مهمې برخې په توګه وظیفه اجرا کوي، او راجسټر(Register) د حافظې (Memory) يو ډ ول دی چې ډېر کم ظرفيت لري. په کمپیوټر کې دوه ډوله حافظه (Memory) استعمالیږي، چې له نیمه هادي (Semiconductormemory) او مرستندويه زېرمې (Hard Disk) څخه عبارت دي. نیمه هادي (Semiconductor Memory) کېدای شي بی ثباته (Volatile) او یا هم باثباته (Non Volatile) اوسي، چې بې ثباته حافظه د برق په بندیدو (Off) سره خپل معلومات له لاسه ورکوي خو با ثباته حافظه د برق په بندیدو (Off) سره بیا هم خپل معلومات ساتي.

۲.۷ نیمه هادي حافظه (Semiconductor Memory)

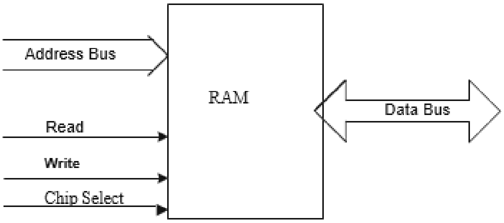
نیمه هادي حافظه (Semiconductor Memory) دوه ډولونه لري:

۱. سملاسي لاسرسي زیرمه (Random Access Memory -RAM)
۲. یوازې لوستونکې زیرمه (Read Only Memory)

په RAM کې ډیټا لیکل(Write) کېدای شي او هم ترې ډیټا لوستل(Read) کېدای شي. څرنګه چې یواځې د برق په موجودیت کې ډیټا ساتي نو په همدې اساس د بی ثباته (Volatile) حافظې په نوم یادیږي.

۱.۷ شکل مونږ ته د RAM عمومی دیاګرام راښايي چې په هغې کې ادرس وصلوونکی (Address Bus) معلوماتو وصلونکی (Data Bus) او د لوستلو (Read) او لیکلو

(Write) سیگنل شامل دی. د معلوماتو وصلونکی (Data Bus) په واسطه ډیټا حافظې (Memory) ته ځي یا هم د حافظې څخه ډیټا ترلاسه کېږي. ادرس وصلوونکی (Address Bus) د حافظې یو ځای (Memory Location) انتخابوي. د لوستلو (Read) سیگنل هغه وخت فعالیږي کله چې د حافظې (Memory) څخه معلومات لوستل(Read) کیږي، او د لیکلو (Write) سیگنل هغه وخت فعالیږي کله چې حافظه کې یو څه لیکل (Write) کیږي. ۱.۷ شکل: 16x8 بیټه د سملاسي لاسرسي زیرمه (RAM) یا 2^4x8 بیټه یا 16 بیټه سملاسي لاسرسي زیرمه (RAM) څرګندوي. بلاک دیاګرام.



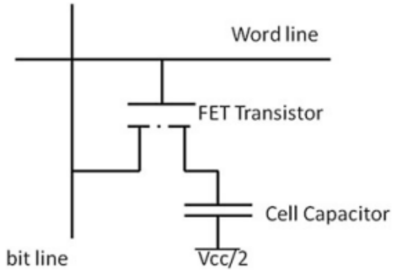
۱.۷ انځور: د Ram بلاک ډیاګرام

Address	
0000	10101011
0001	11001100
0010	10000001
0011	10000000
0100	11111111
0101	11111000
0110	10000000
0111	00000000
1000	00111111
1001	00000001
1010	01010101
1011	10000000
1100	00000011
1101	10000011
1110	11100000
1111	11000000

۲.۷ انځور: د ۱٦ بایټونو رم

Number of addresses	Number of memory locations	Representation
10	$2^{10} = 1024$	1 K
11	$2^{11} = 2048$	2 K
12	$2^{12} = 4096$	4 K
13	$2^{13} = 8192$	8 K
14	$2^{14} = 16,384$	16 K
16	$2^{16} = 65,536$	64 K
20	$2^{20} = 1,048,576$	1 M
24	$2^{24} = 16,777,261$	16 M
32	$2^{32} = 4,294,967,296$	4 G

۱.۷ جدول: د ادرس نمبر او د حافظې موقعیتونه



۳.۷ انځور

په ۲.۷ شکل کې ادرس څلور بیټونه دی نو په همدی اساس $2^4 = 16$ د حافظې موقیعتونه موجود دي . او که چیرې هــر ځای یو B Space ونیسي نو بیا په حافظه کې 16 B راځي. کـه حـافظه m ادرس لینونه (Address Lines) ولري نو بیا ویلی شـو چې 2^m حافظې موقیعتونه موجود دي. ۱.۷ چوکاټ د ادرس لینونو شمیر څرګندوي او هم د حافظې موقیعتونو معادل اعشاري تعداد څرګندوي.

سملاسي لاسرسي زیرمه (RAM) مختلف اقسام یا ډولونه لري لکه:

متحرکه رم یا ډي رم (Dynamic RAM) (DRAM)

هم مهاله متحرکه رم یا ایس ډي رم (Synchronous DRAM) (SDRAM)

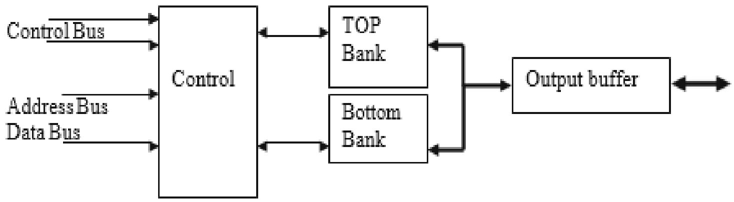
ای ډي او سملاسي لاسرسي زېرمه (EDO RAM) ډي ډي ار ایس ډي سملاسي

لاسـرسـي زېرمه (DDR SDRAM) ساکته سملاسي لاسرسي زيرمه (Static RAM ((SDRAM)

- **متحــرکه ســملاســي لاســرســي زېرمــه Dynamic RAM (DRAM):** دا پـه اصـــلي حـــافـــظه (Main Memory) کې استعمالېږي متحرکه سملاسي لاسرسي زېرمه د يو بيټ جوړولو له پاره يو څو برخې (Components) اســتعمالوي ددی په خاطر چې دا برخې (Components) د مدغم شوی سرکټ (Dram Integrated Circuit) ډيزاين کولی شـي چې ظرفيت يي 4GB Per IC وي. ۳.۷ شـکل: يو بيټ متحرکه سملاسي لاسرسي زېرمه (DRAM) راښايي.

سيل کاپيسيټور (Cell Capacitor) د يو (One) يا صفر (Zero) له منطق (Logic) ســره بـدلیـدلی شـــي خو مګر ديتـه اړتيـا لري تر څو پـه هر يو ملي ثـانيـه (1ms) کې (Refresh) يا تازه شي ترڅو متحرکه رم (DRAM) سرعت کم کيږي.

- **هم مهاله متحرک رم Synchronous DRAM (SDRAM):** د (SDRAM) ټيکنالوجي متحرک رم (DRAM) اوهم د هماهنګۍ له پاره يو خاص مخينه (Interface) استعمالوي چې د دې حافظې د محترک رم (DRAM) په نسبت زيات سرعت لري. SDRAM دوه خپلواکه د حافظې بانکونه (Memory Banks) استعمالوي کله چې يو بانک (Bank) تازه يا بيا فعالېږي. نو د پروسيس مرکزي واحد CPU په بل بانک کې د لوستلو (Read) او ليکلو (Write) عمليات يا فعالېتونه ترسره کوي.



۴.۷ انځور: د SDRAM بلاک دياګرام

۵.۷ انځور: Rambus Memory Module (Courtesy Samsung Coup)

- **(EDORAM) Extended data out RAM ای ډی او سملاسي لاسرسي زیرمه:** ددی په ذریعه د ډیټا بلاک (Block Data) حافظې ته لیرل کیږي یا هم د حافظې څخه نورو برخو ته لیرل کیږي.
- **(DDR SDRAM) Double data rate SDRAM:** دا د SDRAM یو ډول دی چې دا د Clock په Rising Edge او هم په Falling Edge کې ډیټا انتقالوي. دا د SDRAM په نسبت دوه ځلې تیزه ډیټا انتقالوي یعنې سرعت یې د SDRAM دوه برابره دی . نو په همدی اساس حافظه د Clock Rate ½ باندی هم چلیدای یا(Run) کیدای شي . DDR2 او DDR3 د DDR په ټیکنالوجی کې یو پرمختګ دی او وړاندي د ډیټا انتقالولو شمیر زیاتوي. DDR2 RAM په یو دوران کی ۴ ځله د ډیټا لیږد ترسره کوي. او DDR3 په یو دوران (Clock Cycle) کې د ۸ ځله د معلوماتو(8 Data Transfers) لیږد ترسره کوي . د 100 MHZ Clock Rate او معلوماتو وصلونکې (Data Bus) 64 Bits له پاره، د DDR د لیږدونې اندازه (Transfer Rates) عبارت ده له:

DDR = 100x2x8 = 1600 MB/S
DDR2 = 100x4x8 = 3200 MB/S
DDR3 = 100x8x8 = 6400 MB/S

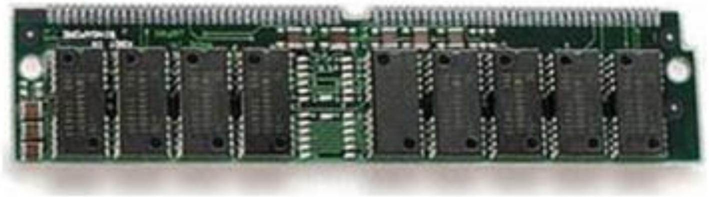
- **(RDRAM) Rambus DRAM ار ډی سملاسي لاسرسي زېرمه:** د رمبوس کارپوریشن (Rambus Corporation) په واسطه رامنځته شوې. او دا د یو نوی مخیني (Interface) سره زیات متحرک DRAM بانکونه استعمالوي متحرک

DRAM بانکونه د دي له پاره فعاله وي تر څومختلف کلمات (Words) انتقال کړي او همدارنگه معلومات د Clock په Rising Edge او Filing Edge کې انتقالوی او د متحرک سملاسي لاسرسي زېرمې (DRAM) بیا فعالېدل (Refreshing) د مخینې (Interface) په واسطه ترسره کیږي. او د متحرک رم (DRAM) دوهم نسل د مستقیم اړدی سملاسي لاسرسي زېرمې (Direct RDRAM) DRDRAM په نوم یادیږي. او ډیټا په 1.6 Gbps کې انتقالولی شي . ۵.۷ شکل: د ار دی سملاسي لاسرسي زېرمې موډل RDRAM Module ښرگندوي.

متحرک رم کڅوړه (Dram Packaging): د متحرک سملاسي لاسرسي زېرمې (Dram) کڅوړه (Packaging) مختلف اقسام لري لکه:

SIMMs (single in-line memory module)
DIMM (dual in-line memory module)

۶.۷ شکل: SIMMs څرگندوي چې یو کوچنی سرکټ تخته (Circuit Board) ده چې ددی تختی په یو طرف کې څو چیپونه (chips) موجود دي. او د معلوماتو 32 بیټه ډیټا بس (Data Bus) لري.



۶.۷ انځور: DRAM SIMM

Memory technology	Access time
SRAM	0.5–2.5 ns
DRAM	50–70 ns
Flash	$5 * 10^3 – 5 * 10^5$

۲.۷ جدول: د حافظې د لاسرسي وخت

DIMM ډی ايي ايم ايم: دا له سـرکټونو جوړه شـوي تخته (Circuit Board) ده چې ددی تختې دواړو خواوو کې زيـات د حـافظې چپونـه Chips موجود دي مګر يو 64bit ډيټا وصلونکې (Data Bus) لري.

(SRAM) Static RAM ثابت سـملاسـي لاسـرسـي زېرمه: دا په کېشـی حافظې (Cache Memory) کې استعمالېږي او ثابت سـملاسـي لاسـرسي زېرمه (SRAM) نسبت متحرک سـملاسي لاسرسي زېرمې (DRAM) ته شل ځلې (Faster) تيز دی او هم يې قيمت زيات دی D Flip-Flop بيا يو بيټ SRAM دی.

يوازې لوسـتونکې زېرمه (ROM): دا هغه حافظه ده چې ددی څخه يواځې معلومات لوسـتل(Read) کېدای شـي. او دا حافظه معلومات د تل له پاره سـاتي .حتی که برق بند (Off) هم وي . او دا حافظه د (Nonvolatile Memory) يا باثباته په نوم هم يادېږي. دوه ډوله ROM لاندی لست شوي:

د پاکيدو او د پروګرام کيدو وړ يوازې لوسـتونکې حافظه (EPROM): دا حافظه د ماورابنفش(Ultraviolet) شـعاع پواسـطه پاکېږي او د يوی الې (Device) پوسـيله دوباره پروګرام (Reprogram) کيږي کوم چې د EPROM Programmer په نوم يـاديږي. FLASH ROM د پاکيدو او د پروګرام کيدو وړ يوازې لوسـتونکې حافظې (EPROM) يو ډول ده.

د بريښـنا پـه ذريعـه د پـاکيـدو او پروګرام کيـدو وړ يوازې لوسـتونکې حـافظې (EPROM): کله چې يو خاص ولتاژ د دې حافظې په يو سوري (PIN) باندی تطبيق (Apply) شـي نو په دې سـره بيا دا حافظه پاکيږي. او دEPROM Programmer پوسيله دوباره پروګرام Reprogram کيږي.

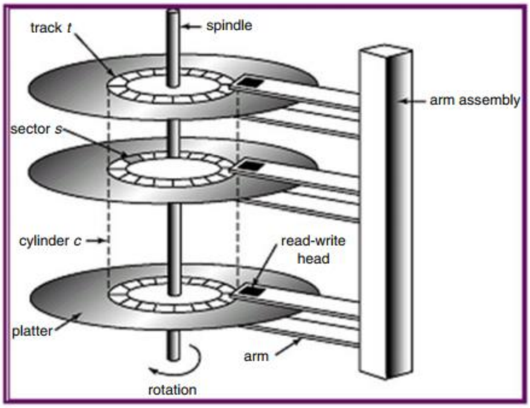
فلش حافظه (Flash Memory): دا يو باثباته حافظه (Nonvolatile Memory) ده چې ډير لوی کاريالونه (Applications) دا حافظې استعمالوي لکه:

Flash Drive, Solid State Drive, Memory Card, Embedded Systems فلش حافظه (Flash Memory) دا د EEPROM یوډول ده چې دی ته اجازه ورکوي تر څو د یوی عملیي (Operation) پواســطه د حافظې مختلف موقیعتونه (Memory Locations) پاک کړي او یا هم پکې ډیټا ولیکل (Write) شـي د فلش حافظې(Flash Memory) له پاره دوه ډوله ټیکنالوجي اسـتعمالېږي: ننډ فلش حافظه (Nand Flash Memory) نار فلش حافظه (NOR Flash Memory). ننډ فلش حـافظه (Nand Flash Memory) د نار فلش حافظی (NOR Flash Memory) پـه نســبـت لږ لاســرســي وخت (Access Time) لري او زیات وخت فلش حافظې (Flash Memories) د ننډ(NAND) ټکنالوژی استعمالوي.

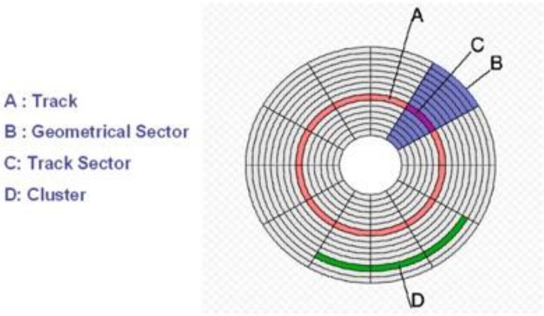
حافظې ته لاســرســي وخت (Memory Access Time): هغه وخت ده چې په هغی کې د پروســیس مرکزي واحد (CPU) د حافظې څخه معلومات لولي(Read) یا هم حافظې کې معلومات (Write) لیکي یا داســې هم ویلی شــو هغه وخت چې په هغه کې د پروسیس مرکزي واحد (CPU) د معلوماتو ادرس وصلوونکې (Address Bus) تـه ورکړي او د هغی پـه اســاس په معلومات وصــلوونکې (Data Bus) کې معلومات ښــکاره شــي او یا هم په حافظې کې معلومات ولیکل (Write) شــي. ۲.۷ چوکاټ: د مختلف قسمه حافظو د لاسرسي وخت(Access Time) څرګندوي.

۳.۷ مرستندویه زېرمه (Hard Disk)

۷.۷ شکل: د مرستندویه زېرمې (Hard Disk) داخلي جوړښت چې د څو پلیټرو (Platters) څخه جوړ شوی او پلیټرونه (Platters) معلومات ذخیره کوي. د هیډ (Heads) وظایف دادي چې د ډیسک (Disk) له سطحې څخه معلومات لولي (Read) یا هم پری لیکې (Write) د پلیټر (Platter) سطحه د څو قابونو(Tracks) څخه جوړه شوې او هر قاب (Track) بیا په په څو حصو (Sectors) باندی تقسیم شوی. لکه په ۸.۷ شکل کې چی ښکاري:



۷.۷انځور: د مرستندویه زېرمې (Hard Disk) داخلي جوړښت



۸.۷انځور: د پلیټر (Platter) سطحه

د ډیسک خاصیتونه (Disk Characteristics)

لاس رسي وخت (Access Time): هغه وخت دی چې په هغه کې د معلوماتو انتقال شروع کیږي او دا وخت د لټون وخت(Seek Time)او څرخیدو ځنډ (Rotational Delay مجموعه ده.

لټون وخت (seek Time): هغه وخت دی چې په هغې کې هیډ (Head) خپل خاص ټریک (Proper Track) ته لیږل (Move) کیږي.

د څرخیدو ځنډ (Rotational Delay): دا ځنډ د څرخیدو په سرعت (Rotation Speed) باندی مربوط دی او د څرخیدو سرعت (Rotation Speed) په Revelation Per Minute (RPM) باندې ښوودل (Represent) کیږي نو په همدی اساس څرخیدو ځنډ (Rotation Delay) په لاندې ډول محاسبه کیږي:

Rotational Delay = Time For Half Revolution = 60s=RPM∗ 2

د ډسک ظرفیت محاسبه کیږی په:

Disk Capacity = Number Of Surfaces∗ Number Of Track=Surface∗ Number Of Sectors=Track∗ Number Of Bytes=Sector

کلسټر (Cluster): د یو ډیسک (Disk) هره حصه (Sector) 512 Bytes لري او کلسټر (Cluster) د یو یا څو حصو(Sectors) څخه جوړ شوی دی که یو کلسټر(Cluster)... 1kb وی نو دا به د دوو حصو (Sectors) څخه جوړ شوی وي یا که کلسټر 2kb وي نو دا به د څلورو حصو(Sectors) څخه جوړ شوی وي. ۳.۷ چوکاټ د کلسټر اندازه او عادی قیمتونه Default Values.... Cluster Size مشخصوي.

۱.۷ مثال: یو ډیسک ډرایو (Disk Drive) اته سطحی لري او هره سطحه 1024 Tracks لري هر ټریک (Track) بیا 64 Sectors حصی لري او هره حصه بیا 512 Bytes او 6000 RPM څرخیدو سرعت (Rotational Speed) لري نو:

a. د ډسک (Disk) ظرفیت څومره دی ؟

b. د ډیسک د څرخیدو سرعت (Disk Rotational Speed) محاسبه کړئ؟

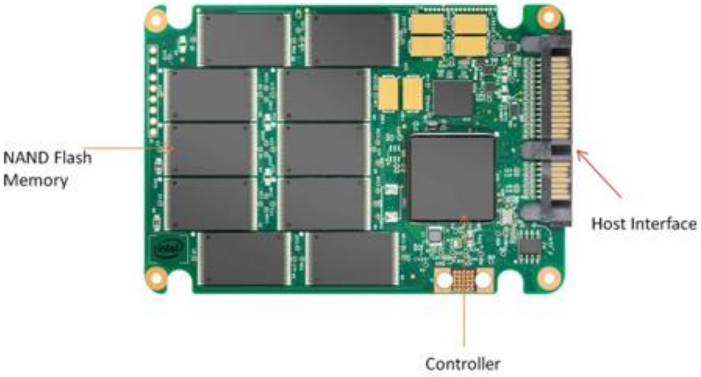
د ډیسک ظرفیت= $8 * 1024 * 64 * 512 = 268{,}435{,}456B$

د څرخیدو ځنډ= $60/600082 = 0.005s$

ډيسک فايل سيستم (Disk File System): فايل سيستم په يو هارډ ډيسک کې د معلوماتو تنظيمول مشخصوي. د پخوانيو وينډوز عامل سيستمونو (Windows OS) له پاره FAT16 او FAT32 فايل سيستمونه استعمالېده خو اوس زياتره وينډوزعامل سيستمونو د Windows OS NTFS(New Technology File System) استعمالوي چې NTFS يي ښه امنيت وړاندې کوي. لکه د لاسرسي (Access) محدوديت او بلی بنې اړولو (Encryption) ته اجازه ورکوي.

DISK size	NTFS cluster size
512–1024 MB	1 kB
1024 MB–2 GB	2 kB
2 GB–2 TB	4 kB

۳.۷ جدول: عادی حالت کې د کلستر اندازه



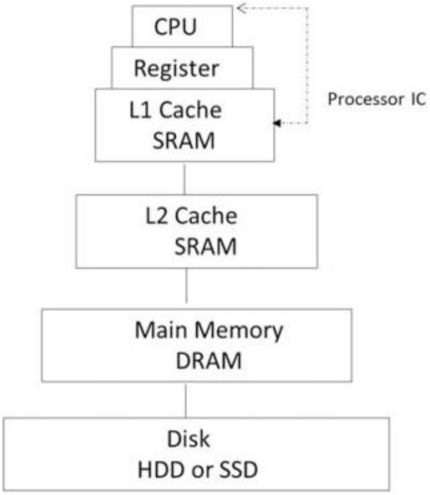
۹.۷ انځور: د ايس ايس ډي (SSD) جوړښت

۴.۷ ایس ایس ډي SSD (Solid State Driver)

هارډ ډیسک یوه پخه اله ده نو په همدی اساس ددې پر ځای اوس زیاتره ایس ایس ډي (SSD) زیات استعمالیږي. ایس ایس ډي (SSD) د باثباته ننډ فلش حافظې (Nonvolatile NAND Flash Memory) څخه جوړ شوی دی. ۹.۷ شکل: د ایس ایس ډي(SSD)جوړښت او ۴.۷ چوکاټ د ایس ایس ډي(SSD) مقایسه د هارډ ډیسک(HDD) سره څرګندوي.

۵.۷ حافظې تسلسل (Memory Hierarchy)

زیاتره کمپیوټرونه دري ډوله حافظو سره راځي چې په مسلسل شکل منظم شوی وي.



۱۰.۷ انځور: د کوچني کمپیوټر د حافظې تسلسل

Memory type	SRAM	DRAM	SSD	HDD
Cost	$8.00/MB	$0.16/MB	$0.20/GB	0.05/GB
Access time	0.5–2.5 ns	50–70 ns	70–150 ns	5–20 ms

۵.۷ جدول: د مختلف قسمه حافظو قیمتونه

۱. **کېشـــي حـافظـه (Cache Memory)**: دا حـافظـه یوه چټکـه حـافظـه ده او اکثره د پروســیس مرکزي واحـد(CPU) په داخل کې اســتعمالېږي چې د L1 Cache په نوم یادیږي. او د اصـلي حافظې په نسبت چټکه ده او په همدی اساس يي د اصلي حافظې په نسبت قیمت هم لوړ دی.

۲. **اصـلي حافظه (Main Memory)**: دا حافظه د متحرک سـم لاسـي لاسـرسـي زېرمه (DRAM) اوایس ډی سـم لاسـي لاسرسي زېرمه (SDRAM) استعمالوي کله چې یو پروګرام چلیږي نو هغه باید د مرسـتندویه حافظې یعنی هارډ ډیسـک څخه اصـلي حافظې (Main Memory) ته راوړل شي.

۳. **(Secondary Memory) دایمی حـافظـه**: لکه : SSD, HDD, CD-ROM ( ) ۵.۷ چوکاټ

کېشی حافظه (Cache memory)

د کېشـــی حافظه (cache memory) هرځای دکېشـــی حافظې لین (Cache Line) په نوم یادیږي چې هر لین (Line) دکېشـی حافظې (Cache Memory) نه یو بلاک دمعلوماتو را انتقالوي. د پروسـیس مرکزي واحد (CPU) کې دوه دوه ډوله کېشی حافظې موجودې دي:

۱. **ډیټا کېشـی حافظه (Data cache)**: دا کېشـی حافظه (Cache) معلومات سـاتي چې دا معلومات بیا د د پروســیس مرکزي واحد (CPU) پوســیله لوســتل (Read) او لیکل (Write) کېدای شي.

۲. **هـدایـات پـه کېشــی حـافظـه کې (Instruction Cache)**: دا لارښـــوونې (Instruction) یا هدایات ساتي اود پروسیس مرکزي واحد (CPU) یواځې د ال یو کېشی حافظې (L1 Cache) نه دا لوستل (Read) کولی شي.

د کېشي حافظې اصطلاحات (Cache Terminology):

پاتی کېدنه (Miss): کله چې پروسيسر د معلوماتو له پاره کېشي حافظې څخه غوښتنه وکړي خو دغه معلومات له کېشي حافظی سره موجود نه وي نو دي ته کېشي حافظې پاتی کېدنه (Cache Miss) وايي.

توافق کول (Hit): کله چې پروسيسر د معلوماتو له پاره کېشي حافظې څخه غوښتنه وکړي او کېشي حافظې سره دغه معلومات موجود وي او پروسيسر ته يې ورکړي نو ديته کېشي حافظې توافق (Cache Hit) وايي.

توافق نسبت (Hit Ratio): کله چې پروسيسر د معلوماتو له پاره کېشي حافظې څخه غوښتنه کوي که چيري څو ځلې پاتي راشي نو د هغې شمير ته توافق نسبت ته ويل کيږي يا هم د لوستلو مجموعې شمير ته توافق نسبت ويل کيږي.

بلاک (Block): څو کلمات چې سره يو ځای شي يو بلاک جوړوي يا په بل عبارت د اصلي حافظې هر يوه ځای ته بلاک ويل کيږي.

فزيکې ادرس (Physical address): هغه ادرس چې د پروسيس مرکزي واحد (CPU) د هغې په واسطه اصلي حافظې ته لاسرسي پيداکوي.

خيالي ادرس (Virtual address): خيالي ادرس هغه ادرس دی چې د پروسيس مرکزي واحد (CPU) د هغې په واسطه خيالي حافظې يا دايمي حافظې ته لاسرسی پيدا کوي.

کېشي لين او کېشي بلاک (Cache line or cache block): په هرلين (Line) يا هربلاک (Block) کې مختلف کلمات يا بايټس زېرمه کيدای شي د کېشي حافظې کې دحافظې لين (Cache Line) اندازه په اصلي حافظه کې د بلاک له اندازی سره مساوي وي.

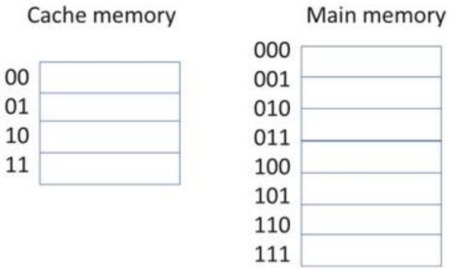
د کېشــی حـافـظې د کـړنلارو مـیـتودونـه (Cache memory mapping methods):

۱۱.۷ شــکل: یو کېشــی حافظه ښــایي چې څلور موقعیتونه لري او هم اصــلي حافظه اته موقیعتونه لري نو پدې صــورت کې کېشــی حافظه د اصــلي حافظې د څلورو ځایونو معلومات زېرمه کولی شي. د پروسیس مرکزي واحد لومړی کېشي حافظې ته لاس رسی پیدا کوي که کېشی حافظه کې معلومات نه وي نو بیا اصـلي حافظې څخه غوښـتنه کوي چې بیا یو بلاک معلومات د اصـلي حافظې څخه کېشــی حافظې ته لیږل کیږي او د پروسـیس مرکزي واحد(CPU) ته لاس رسی پیدا کیږي. پوښتنه داده چې دا معلومات به د کېشـی حافظه په کوم ځای کې زېرمه کیږي؛ چی دا موضوع د کیشی حافظې د کړنلارو میتودونه بحث منځ ته راوړي.

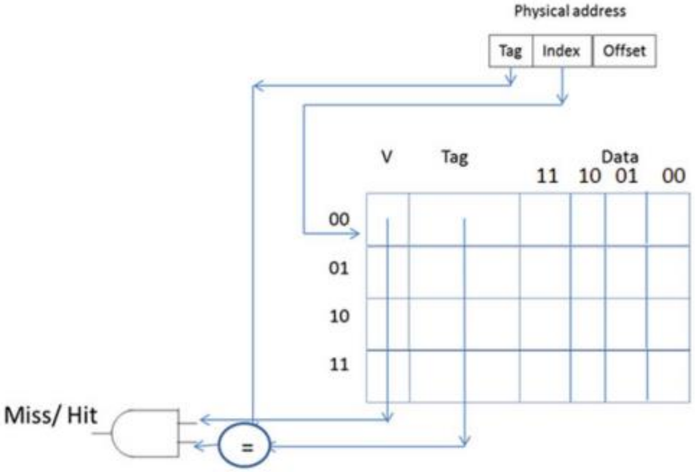
نو ویلی شــو چې د کړنلارو میتود هـدف دادی چې د اصـــلي حـافظې څخـه چې کوم معلومات کېشـی حافظې ته راځي نو د اصلي حافظې کوم بلاک باید په کېشی حافظه کې ځـای پرځای (Replace) شــي مطلب د بلاک معلومات باید د کېشـــی حافظې په کوم لینونو کې ذخیره شي. لاندي د میتودونو هغه ډولونه دی چې د کړنلارو له پاره استعمالیږي:

۱. مستقیمه کړنلاره (Direct Mapping)
۲. پیوسته کړنلاره (Associative Mapping)
۳. د پیوسته کړنلاری سیټ بندي (Set Associative Mapping)

مســتقیمه کړنلاره (Direct mapping): په دی کړنلاره کې د اصــلي حافظې یو بلاک راخیســـتل کیږي او د کېشـــی حافظې په یو خاص لاین کې ځای پرځای کیږي چې دغه لاین پـه حقیقیـت کې د همـدغـه بلاک له پاره خاص شـــوی وي. د مســـتقیمی کړنلاری غوښتنه دا ده چې د حافظې صفر بلاک (Block 0) د کېشی حافظې په صفر (Line 0) کې ځای پر ځای شی. یعنی پدې کې د اصلي حافظې بلاک باید د کېشی حافظې د لاین سـره مسـاوي وي. ۱۲.۷ شـکل یو کېشـی حافظه ښایي چې څلور لینونه ولري او هر لاین څلور بـایتـه معلومـات زېرمـه کوي او د معلومـاتو فزیکي ادرس پـه درې کـالمونو بـاندی تقسیموي.



۱۱.۷ انځور: کیشی او اصلي حافظه



۱۲.۷ انځور: کېشی حافظه چې څلور کېشی لاینونه لري.

انډولول(Offset): دا د کېشـــی حافظې لاین اندازه مشـــخصـــوي چې په کېشـــی حافظې لاین کې څومره بـایټس معلومـات موجود دي. څرنګه چې د کېشـــی هر لاین ۴ حافظې موقعیتونه ساتلی شي نو انډولول (Offset) یې دوه بیټه دي همدارنګه انډولول مشخصوي چې کوم یو په څلورو معلوماتو کې د پروسیس مرکزي واحد ته ولېږدول شي.

شـاخص (Index): دا د کېشی حافظې ادرس مشخصوي. صحیح بیټ V-Bit (Valid Bit) دا بیټ په یو سـره ښـودل کیږي او دا مشـخصـوي چې ایا د کېشـی حافظې لاین

معلومات صحیح دی یا نه.

ټیګ (Tag): د ټیګ اندازه مساوي ده له:

(د انډولولو اندازه (Offset+ د شاخص اندازه) - د فزیکي ادرس اندازه

که چېرته فزیکي ادرس 7 بیټه وي نو د ټیګ اندازه عبارت ده له:

$$7 - (2 + 2) = 3$$

که چېرته صحیح بیټ یو وی د ادرس فیلډ ټیګ له هغه ټیګ سره چې په کېشی کې موجود دی برابر وی نو نتیجه یي Hit ده پرته له دی یي نتیجه Miss ده.

کله چې د پروسیس مرکزي واحد له کېشی څخه Miss تر لاسه کړی نو اصلي حافظې ته لاسرسي پیداکوي او د معلوماتو بلاک له اصلي حافظې څخه د کېشی لینونو ته په مستقیمې کړنلاری کې د لاندې معادلې په استعمال سره لیږدوي.

Cashe Line Address=(Main Memory Block Number) Module N

چې N د کېشی د لینونو نماینده ګی کوي.

۱۳.۷ شکل په نظر کې ونیسئ فرض کړئ د پروسیس مرکزي واحد 11 10 010 ادرس تولیدوي ۱۳.۷ شکل د کېشی له خوا لیدل شوی فزیکي ادرس بڼه ښکاره کوي.

د پروسیس مرکزي واحد د کېشی Line10 ته د لاس رسی له پاره د شاخص قیمت استعمالوي که چېرته صحیح بیټ صفر وي نو نتیجه یي Miss ده د پروسیس مرکزي واحد اصلي حافظې ته لاسرسي پیداکوي او 01010 بلاک د کېشی لسم لاین د لاندې معادلې له مخې لیږدوي. د مستقیمی کړنلاری مشکل دادی چې په دې کې باید د اصلي حافظې څخه مکمل بلاک کېشی حافظې ته لاړ شي یعنی داسې نه شي کیدای چې د اصلي حافظې یو لوی بلاک د کېشی حافظې په لاین کې ځای پر ځای شي چې ددی مشکل د حل له پاره د پیوسته کړنلارې میتود منځته راغلی.

Tag	Index	Offset
010	10	11

۱۳.۷ انځور د 0101011 فزیکي ادرس جوړښت

	V	Tag	Data 11	10	01	00
00	0					
01	0					
10	1 ~~0~~	010	M(2B)	M(2A)	M(29)	M(28)
11	0					

۱۴.۷ انځور کېش حافظه د څلور لینونو (څلورو بلاکونو سره)

(01010) 2 = (10)10
Cache line number = 10modulo 4 = 2 or (10) 2

په دی حالت کې هر بلاک د څلورو حافظې موقعیتونو څخه جوړ شوي دي نو انډول یي دوه بیټه دی دا کولی شي هر یو $4(2^2)$ د معلوماتو برخو ته د حافظې د بلاک په داخل کې مراجعه وکړي

۰۱۰۱۰ ادرس باندی بلاک دوه بیټه انډول سـره ۰۱۰۱۰۰۰ (M28), 0101001 (M29), 0101010 (M2A), And 0101011 (M2B) حـافظې موقیعتونـه لري.

په دی حالاتو کې M28 M29, M2a, او M2b د کېش لسـم لاین ته لېږدول کیږي او صـحیح بیټ یي یو کیږي ۴.۷ شـکل کې ښـودل شوي چې د ټیګ د کېش ټیګ فیلډ کې ځای پر ځای شوی دی.

اوس کـه چېرتـه د پروســیس مرکزي واحـد ۰۱۰۱۰۱۰ ادرس تولیـد کړی چېرتـه چې Tag=010، Index=10 او Offset =10 د پروســیس مرکزي واحـد شــاخص اسـتعمالوي تر څو لسـم لاین ته لاسـرسـي پیداکړي که چېرته یو کېش لاین لسـم وي نوصـحیح بیټ یي یو دی او د ادرس ټیګ په کیشـی لاین کې له موجود ټیګ سـره برابر دی نو نتیجه یي Hit ده او د پروسیس مرکزي واحد لسم انډول (Offset) استعمالوي تر څو (M2A) د پروسیس مرکزي واحد ته ولېږدوي.

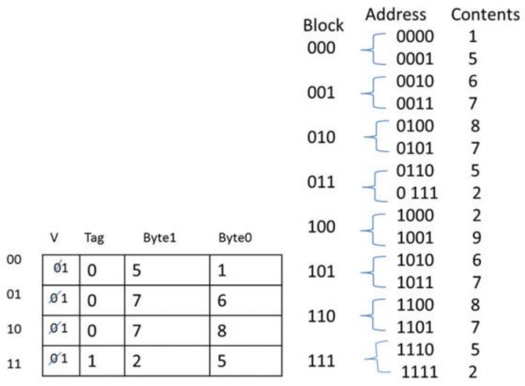
(Where Offset 00 = M28, 01 = M29, 10 = M2a, 11 = M2B)

مثال ۲.۷: شـکل ۱٥,۷ د کمپیوټر اصـلي حافظه او کیشی حافظه څرګندوي د پروسیس مرکزي واحـد د ۱٦ پـه قـاعده 0 × 0,0 × 2,0 × 5 and 0 × 2. ادرس جوړوي.

فرض کړئ په لومړی سر کې کیشی حافظه خالي ده تاسو د کیشی محتویاتو وښایئ.

پـه دی مثال کې هر بلاک په کیشـــی لاین کې 2b دی او اصــــلي حافظه اته بلاکونه لري ۱٦.۷ شکل د کیشی حافظې له خوا لیدل شوی فزیکي ادرس څرګندوي.

- د بایټ انډول (Offset) یو بیټ دی (2 B Per Block)
- شاخص (Index) دوه بیټه دی
  (Cache Consist Of 4 Lines $=2^2$)
- ټیګ (Tag) یو بیټ دی
  (Bits In Block - Bits In Index = 3 – 2=1)
- د پروســیس مرکزي واحد ۰X۰ یا 0000 ادرس جوړوي دهمدی له پاره ټیګ صـفر انډکس 00 او انډول (Offset) یي صـفر دی د پروسـیس مرکزي واحد کیشـی لایدن 0000 ته لاسرسـي پیداکوي او صـحیح بیټ یي صـفر دی چې نتیجـه یي Miss ده د پروســـیس مرکزي احد د اصـــلي حافظې 0000 ادرس ترلاســـه کوي او000 بلاک 00 لاین تـه یي لېږدوی او د ټیـګ بیټ صــــفر او دصحیح بیټ یو وي



۱٥.۷ انځور د ۲.۷ مثال کیشی او اصلي حافظه

1	2	1
Tag	Index	Byte offset

۱٦.۷ د ۱.۷ مثال له پاره فزیکي ادرس

د دي له پاره کیشــی 00 لاین $V = 1, \text{Tag} = 0, \text{Byte1} = 0 \times 5, \text{Byte0} = 1 \times 0$ لري.

- راتلونکې د پروسـیس مرکزي واحد 0x2 یا 0010 ادرس جوړوي شـاخص یي 01 دی له دې کبله د پروسـیس مرکزي واحد 01 د کیشــی لاین ته لاسـرســي پیداکوي چېرته چې صــحیح بیټ صــفر دی نو نتیجه یي Miss دی نو د پروســیس مرکزي واحد د اصـلي حافظې 0010موقعیت ته لاسرسي پیداکوي او ۰۰۱ بلاک د کیشی ۰۱ ته لیږدوی صحیح بیټ یي یو کیږي او د ادرس ټیګ برخه د کیشی په لاین کې ځای پر ځای کوي ددی له پاره د 01 د کیشی لاین به اوسي $V = 1, \text{Tag} = 0, \text{Byte1} = 0 \times 7, \text{Byte0} = 0 \times 6$ وي
- د پروسـیس مرکزي واحد 0x5 یا0101 ادرس تولیدوی شـاخص یي 10 دی له دی کبله د پروســیس مرکزي واحد 10 د کیشــی لاین ته لاســرســي پیداکوي چېرته چې صـحیح بیټ صـفر دی نو نتیجه یي Miss دی نو د پروسیس مرکزي واحد د اصـلي حافظې0101 موقعیت ته لاسرسي پیداکوي او 010 بلاک کیشی ته لیږدوی صحیح بیټ یي یو کیږي او د ادرس ټیګ برخه د کیشی په لاین کې ځای پر ځای کوي ددی لـه پـاره د 10 د کیشــی لاین بـه اوســي $V = 1, \text{Tag} = 0, \text{Byte1} = 0 \times$ $7, \text{Byte0} = 0 \times 8$ وي

د پروسیس مرکزي واحد 0x2 یا 0010 ادرس جوړوي د پروسیس مرکزي واحد 01 د کیشی لاین ته لاسرسي پیداکوي چېرته چې صحیح بیټ یو دی او دکیشی لاین ټیګ د ادرس له ټیګ سره برابر دی کوم چې نیتجه یي Hit دی اهډول (Offset) یي صفر دی

نو د پروسیس مرکزي واحد د کیشی لاین 01 څخه 0 بایت لولي دغه پروسه د نورو ادرسونو له پاره همدا ډول دوام کوي

	V	Tag	Byte1	Byte0
0	0			
1	0			

V	Tag	Byte1	Byte0
0			
0			

۱۷.۷ انځور: دوه اړخیزه ترتیب شوی کیشی

2	1	1
Tag	Set	Byte Offset

۱۸.۷ انځور د کېشی په واسطه لیدل شوی د فزیکې ادرس جوړښت د پیوسته کړنلارې لپاره

پورتنی د کیشی او اصلي حافظې د یو شان اندازي په استعمال سره 000 او 100 دواړه بلاکونه به د کیشی 00 لاین ته (Map) اړول کیږي که چېرته د پروسیس مرکزي واحد 0000,1000,0001 او 1001 په مسلسل ډول تولید کړي د څلورو واړو ادرسونو له پاره به نتیجه Miss وي د دي له پاره چې Miss کم کړو نو کیشی به په سیټونو تقسیم کړو چې دی کړنلاری ته پیوسته کړنلاره وایي.

پیوسته کړنلاره (Associative mapping)

په دي کړنلاره کې د اصلي حافظې څخه چې هر بلاک یا هر څومره معلومات کېشی حافظې ته ځي نو د کېشی حافظې په هر لاین کې ځای پر ځای کیدای شي . مثلا که په اصلي حافظه کې یو بلاک 48-bit وی او د کېشی حافظې یو لاین 32-Bit وی نودا 48-Bit معلومات به بیا په دوو برخو ویشل کیږي چې ددی معلوماتو 32-Bit په یو لاین او باقی بیټونه یي په بل لاین کې ذېرمه کیږي.

د پيوسته کړنلارې سيټ بندي Set associative mapping

دا هغه کړنلاره ده چې په دی کې د مستقيمی کړنلارې او پيوسته کړنلارې ښه مشخصات سره يو ځای شوی په دی کې کېشی حافظه په سيټونو باندی تقسيم کيږي او د هر سيټ اندازه پکې تغير کيدای هم شي.

مثالونه : پدې کې Two-Way Set Associative او Four-Way Set Associative راتلی شی. ۱۷.۷ شکل Two-Way Set Associative Cache او ۱۸.۷ شکل د کېش په واسطه ليدل شوی د فزيکې ادرس جوړښت ښرگندوي. د تير مثال په استعمال سره بايټ انډول (Byte Offset) به يو بيټ پاتي شي او دا به په کيشی حافظه کې موجوده دوه دوه بايټونو څخه يو ته رجوع کوي د سيټ تشخيص به يو بيټ وي چې د سيټ ادرس تعريف کړي. د ټيگ اندازه په لاندې ډول محاسبه کيږي.

Tag Size = Physical Address Size – Set Address - Offset

فرض کړی چې فزيکې ادرس ۴ بيټه دی نو ۱۸.۷ شکل د کيشی په واسطه ليدل شوی فزيکې ادرس ښرگندوي.

۳.۷ مثال: ۱۵.۷ شکل اصلي حافظه په پام کې ونيسئ چې کيشی حافظه خالي او په دوو سيټونو باندی تقيسيم شوې ده لکه څنگه چې په شکل ۱۹.۷ کي ښودل شوي که چېرته د پروسيس مرکزي واحد

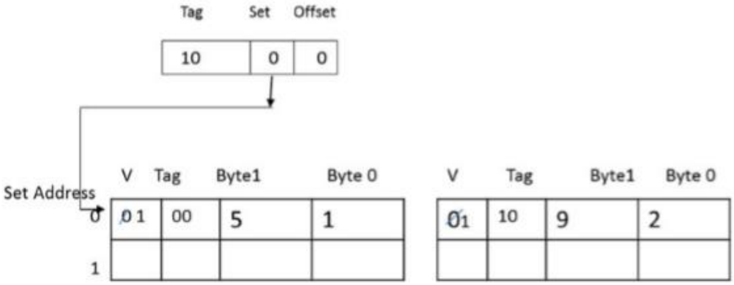
8×0 and $0 \times 8{,}0 \times 0{,}0 \times 0$ادرسونه توليد کړي نو د کيشی محتويات ښکاره کړئ.

د پروسيس مرکزي واحد 0x0 يا 0000 ادرس جوړوي او د کيشی 0 سيټ ته لاسرسي پيدا کوي دواړه د کيشی لينونه په 0 سيټ کې د 0 صحيح بيټونه لري له دی کبله نتيجه يي Miss ده د پروسيس مرکزي واحد بيا اصلي حافظې ته لاسرسي پيدا کوي د حافظې 0000 او 0001 موقعيتونو محتويات کيشی ته لېږدوی نو صحيح بيټ ی و ته بدليږي او

د ادرس ټېګ برخه په کېشی کې ځای پر ځای کیږي.

راتلونکي کې د پروسیس مرکزي واحد 0x8 یا 1000 ادرس جوړوي او کیشی سیټ ته لاسرسي پیدا کوي لومړی لاین د کیشی سم یو بیټ لري لیکن ټیګ ورسره برار نه دی (10vs00) . د پروسیس مرکزي واحد اصلي حافظې ته لاس رسی پیدا کوي 1000 او 1001 د حافظې موقعیتونو محتوا د کیشی د 0 سیټ دوهم لاین ته لېږدوي او صحیح بیټ یي یو ته تبدلیږي او د ادرس ټیګ برخه د کیشی په ټیګ کې ځای پر ځای کوي اوس کله چې د پروسیس مرکزي واحد 0x0 او 0x8 ادرس جوړ کړی بیا یي هم نتیجه Hits ده.

۲۰.۷ انځور: ۰۰۰۰ او ۱۰۰۰ له پاره د کیشی محتویات



۱۹.۷ انځور: د 0000 ادرس له پاره د کیشی محتوا

د بدلولو کړنلاره (Replacement Method)

په پیوسته کړنلاره کې کله چې مرکزي پروسیس واحد د معلوماتو یو نوي بلاک کیشی ته راوړي، نو یو لین د کیشی باید حتماً د نوي بلاک سره بدل شي.

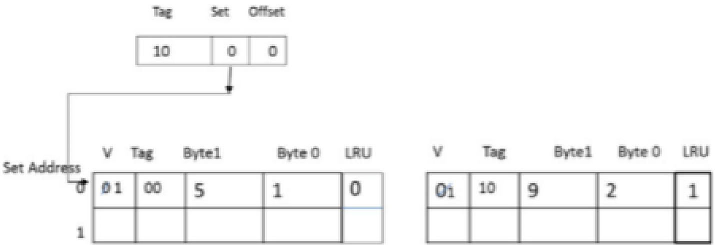
۲۰.۷ شکل په نظر کې ونیسئ که چېرته د پروسیس مرکزي واحد:

$$0100(4)(tag = 01, set\ address = 0, and\ offset\ is = 0)$$

ادرس جوړ کړي دا 0 سیټ ته لاس رسي وکړي او دواړه د کیشی لاینونه یو سم بیټ لري لیکن په کیشـی لاین کې ټیګونه د ادرس له ټګ سره برابر نه دي نتیجه یي Miss ده او د

پروسيس مرکزي واحد اصلي حافظې ته لاسرسي پيدا کوي د ۰۱۰۰ او ۰۱۰۱ حافظې موقعيتونو محتوا بايد کيشى ته ولېږدوي. لکه څنګه چې په ۲۱.۷ شکل کې څرګند شوي

په ۲۱.۷ شکل کې فرض کړئ په سيټ 0 دواړه د کيشى لينونه خالي دي يو نوى بلاک په سيټ 0 کې د لمړي کيشى ته لېږدول کيږي او LRU له صفرڅخه يو ته تبدليږي دوهم بلاک د سيټ 0 کې دوهم کيشى لين ته ليږدول کيږي او LRU له صفر څخه يو ته تبدليږی مگر په يو وخت کې دا کار صورت نه نيسي LRU لمړی کيشى لين له يو څخه صفر ته بدلوي نو ددی له پاره د کيشى لين د LRU=0 سره هغه بلاک ولري چې په کيشى لاين کې د اوږد مهال له پاره وي.



۲۰.۷ انځور

Valid	Address	Byte1	Byte0
1	*000*	5	1
1	*100*	9	2
0			
0			

۲۱.۷ انځور: پيوسته کړنلاره

کمله پیوسته کړنلاره (Fully associative mapping)

په مکمله پیوسته کړنلاره کی داخلي ادرس د ډیټا سره یو ځای په کیشی حافظه کي زېرمه کیږي. ۲۲.۷ شکل یو مکمله پیوسته کیشی د څلورو لینونو سره څرګندوي کله چی مرکزي پروسیس واحد اصلي حافظه د
0 *0 او 8 * 0 موقیعت ته لاس رسی پیدا کړي. که چیري مرکزي پروسیس واحد پیوسته 0 * 0یا 0000 ادرس جوړ کړي، نو دا ادرس به له هر یوه ادرس سره په کیشی کی مقایسه کړي، که چیري یی سمون وخوړ، نو به بیا له کیشی څخه ډیټا ولولي..

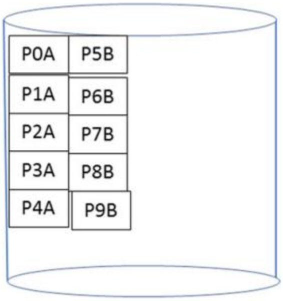
د کیشی حافظه نوې کول (Cache update) میتودونه:

۱. *د لیکلو په واسطه (Write Through)*: که پروسیسر په کومومعلوماتو کې تغیر رامنځته کړی او بیا یي کېشی حافظې ته ورکړي نو دغه تغیرات مستقیما په اصلي حافظه کې هم لیکل (Write) کیږي چې په دې سره دواړه ځایونو کې معلومات نوي کیږي.

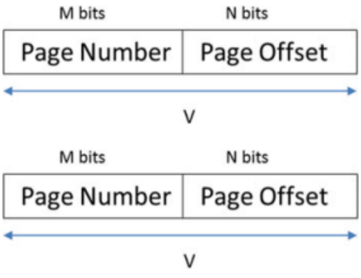
۲. *د بفر لیکلو په واسطه (Buffered write through)*: د کیشی حافظې او اصلي حافظې تر منځ د بېر حافظه واقع ده نو کله چې نوي معلومات کېشی حافظې کي لیکل (Write) کیږي نو بفر کې هم لیکل کیږي چې په دې صورت کې بیا د پروسیس مرکزي واحد (CPU) د بفر حافظې ته لاس رسی پیداکوي یعنی مخکې له دی نه چې دا معلومات اصلي حافظې کي ولیکل (Write) شي نو د پروسیس مرکزي واحد (CPU) یي د بېر نه ترلاسه کوي.

۳. *د کېشی په زیرمه شوو معلوماتو کې تغیرات راتلل (Write back)*: پدې طریقه کې یوازې د کیشی معلومات نوي کیږي ،اود اصلي حافظې معلومات هغه وخت نوي کیږي کله چې د مربوطه کیشی لاین کې نوي معلومات راشي.

پدي طريقه کې دهر کیشـی لاین ډيرټي (Dirty Bit) لري، او ددې پوسـیله معلوميږي چې کيشی لاين تغير شوی يا نه.



۲۲.۷ انځور: خيالي حافظه



۲۳.۷ انځور: خيالي حافظي ادرس بڼه

۱.۵.۷ خيالي حافظه (Virtual Memory): خيالي حافظه HDD يا SSD وي، دا د کاريالونو د هغي ډيټا او لارښوونو د زېرمه کولو له پاره استعماليږی کومي چې د پروسيس له پاره د مرکزي پروسيس واحد له پاره فی الحال په کار نه وي. خيالي حافظه هغه حافظه ده چې متحرک سم لاسی لاسرسي زېرمې (Dynamic RAM) له پاره د هارډ ديسک يوه برخه نيسي يا ريزرف کوي او د سم لاسی لاسرسي زېرمې (RAM) سره دمعلوماتو په وړاندې کولو کې مرسته کوي او دې حافظې ته ځکه خيالي حافظه وايي کله چې معلومات يا پروګرام د چليدو حالت څخه وځي نو دا حافظه بيرته له منځه ځي او د هارډ ډيسک يوه برخه ګرځي. مثلا که سم لاسي لاسرسي زېرمه (RAM) دوه 2 GB وي

خو 8GB معلومات چلولی (Run) شی نو په اصل کې 2GB معلومات په رم (RAM) کې وي او نوره 6GB معلومات په خيالې حافظې کې وي. خيالې حافظه يو ايس ايس ډی هارډ ديسک ده او د دی له پاره استعمالېږي تر څو هغه د کارېال معلومات او لارښوونې زېرمه کړي چې هغه همدا مهال د پروسيس وي او د مرکزي واحد CPU پواسطه لاسرسي نه ورته کيږي يا هم د هغی پروسيس ته ضرورت موجود نه وي. خيالې حافظه يو سيستم ته د دې اجازه ورکوي تر څو د اصلي حافظې په نسبت لوي کاريالونه وچلوي که د پروسيس مرکزي واحد 16 Address Lines ادرس لينونه ولري نو د خيالي حافظې اندازه به 2^{16} بايټه وي. خيالي حافظه په حصو باندی تقسيمېږي لکه څنګه چې په ۲۳.۷ شکل کي ښکاري؛ پروسيس A د P0 نه تر P4 صفحې ريزرف کوي او پروسيس B له P5–P9 پوري ريزرف کوي.

د پروسيس مرکزي واحد په ډيسک کې د يو ادرس يا معلوماتو د لاسرسي پيدا کولو له پاره د V-bits يو خيالي ادرس جوړوي چې دا بيټونه بيا په دوو شاخصونو (Identifiers) باندي ويشل شوي وي. لکه په ۲۴.۷ شکل کې چې ښکاري. په يو سيستم کې د صفحو مجموعه له 2^M سره مساوي ده او په يو ه صفحه کې د بايټونو تعداد له 2^N سره مساوي ده او د ادرسونو مکمل شمير له $2^{(NþM}$ يا هم 2^v سره مساوي ده.

۴.۷ مثال: د يو خيالي ډيسک ظرفيت 2MB ده او هره صفحه...2KB دی البته په يو ادرس ورکونکي سيستم کې.

a. د N او M قيمتونه پيدا کړئ؟

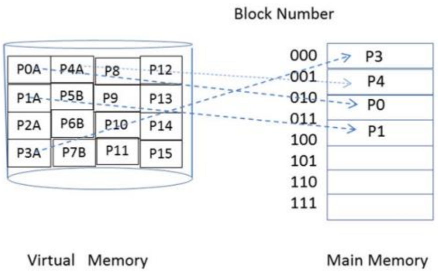
b. په ډيسک کې څومره صفحې موجودې دي؟

Since each page is 2 kB, $2^N = 2048$ B, meaning N equals 11. The capacity of the disk is 2 MB. $2^V = 2$ M equals 2^{21} so V = 21 bits.

نو د صفحو تعداد له $2^{(21-11)}$ سره مساوي دی او په ډيسک کې 2^{10} يا هم 1024صفحې

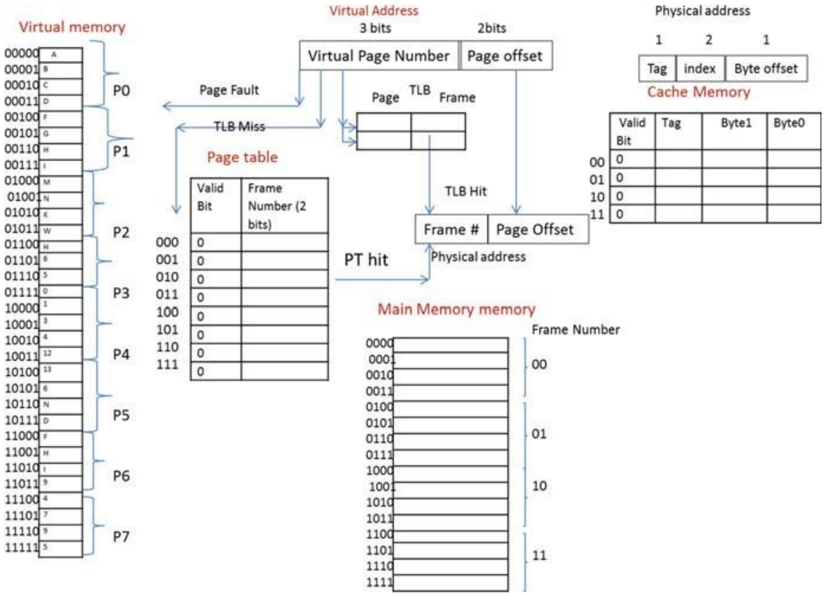
موجودي دي او د هرې صفحې اندازه له 2KB سره مساوي ده.

د صفحو جدول (page table): د اصلي حافظې سره يو ځای په بلاکونو باندې تقسيميږي او د هر بلاک اندازه د صفحې له اندازې سره مساوي وي. کله چې د پروسيس مرکزي واحد يوه صفحه حافظې ته ليږي نو د اصلي حافظې د صفحو جدول (Page Table) کې د هغو صفحو نمبر او بلاک له ځان سره ساتي. د صفحو ادرس لاين په خپله صفحه نمبر دی او هر لاين په اصلي حافظه کې د مقايسه شوي ځای بلاک نمبر او يوصحيح بيټ له ځان سره لري چې دا مشخصوي چې ايا دا لاين صحيح دی يا نه. ۲۵.۷ شکل د صفحو جدول مشخصوي چې په هغې کې n صفحې موجودي دي P0, P1, P2, P3 او P4 چې بلاک ته ليږل کيږي.



Page Number	V	Block Number
0000	1	010
0001	1	011
0010	0	
0011	1	000
0100	1	001
0101	0	
0110	0	
0111	0	
1000	0	
1001	0	
1010	0	
1011	0	
1100	0	
1101	0	
1110	0	
1111	0	

۲۴.۷ انځور: خيالي حافظه



۲۵.۷ انځور: د کمپیوټر د حافظې تنظیمونکی

هر پروسیس په حافظې کې خپل د صفحو جدول لري نو په همدې اساس کېشي حافظه تیزه ده او د صفحو جدول یو ه برخه چې Translation Lookaside Buffer (TLB) په نوم یادیږي دا په کېشي حافظه کې زېرمه وي او TLB د Associative Mapping استعمالوي.

۲.۵.۷ د کمپیوټر د حافظې جوړښت (Memory Organization Of A Computer)

۲۶.۷ شکل د کمپیوټر د حافظې جوړښت روښانوي چې په هغې کې لاندې برخې شاملې دي:

A. خیالي حافظه(هارډ ډسک یاکلک حالت لرونکی ډرایو).

B. اصلي حافظه (د A type of DRAM).

C. کېشي حافظه (SRAM).

D. د صفحو جدول کوم چې په اصلي حافظه کې د صفحو ځرک معلوموي.
E. ټی ایل بی (TLB) کوم چې د صفحو جدول یوه برخه له ځان سره ساتي.

د حافظې عملیې (Memory Operation)

دا لاندېني مراحل د کمپیوټر د حافظې عملیې یا فعالېتونه مشخصوی چې لومړی د پروسیس مرکزي واحد (CPU) یو خیالي ادرس جوړوي او TLB یي چېک کوي چې ایا مربوطه صفحه په اصلي حافظه کې له مخکې نه موجوده ده یا نه.

A. که چیري TLB وښايي چې مربوطه صفحه په اصلي صفحه کې موجوده ده نو بیا یو فزیکې ادرس جوړوي او معلومات په کېشی حافظه کې چېک کوي:

a. که معلومات په کېشی حافظه کې موجوده وی نو دا حالت د هیټ په نوم یادیږي او پدې صورت کې معلومات د کېشی حافظه څخه لوستل (Read) کیږي.

b. که معلومات په کېشی حافظه کې موجوده نه وی نو دا حالت د میس په نوم یادیږی او پدې صورت کې د پروسیس مرکزي واحد حافظې ته لاسرسی پیداکوي او د اصلي حافظې څخه بلاک معلومات کېشی حافظې ته انتقالوي، او بیا د صفحو جدول نوي کوي، TLB نوي کوي او دا پروسه بیا له سره کوي.

B. که چیري مربوطه صفحه په TLB کې نه وي نو د پروسیس مرکزي واحد بیا صفحو جدول پلټي (Check):

۱. که مربوطه صفحه په اصلي صفحه کې وی نو بیا TLB نوي (Update) کوي او د لومړی قدم یا مرحلی یي نه یي دوباره تکراروي.

۲. که مربوطه صفحه په اصلي حافظه کې موجوده نه وي نو بیا صفحه د خیالي حافظې څخه اصلي حافظې ته انتقالوي (Move) او بیا د صفحو جدول او TLB نوي (Update) کوي او دوباره یي د لومړی مرحلې څخه تکراروي.

C. که مربوطه صفحه د صفحو په جدول کې نه وي نو بیا د پروسیس مرکزي واحد مربوطه صفحه د خیالي حافظې څخه اصلي حافظې ته انتقالوي (Move) بیا د صفحو جدول او TLB نوي (Update) کوي او بیرته یي د لومړی مرحلی (Step) نه یي تکراروي.

پوښتنې او ستونزې

۱. د بې ثباته (Volatile) او باثباته (Nonvolatile) حافظو تر منځ فرق په څه کې دی؟

۲. سم لاسی لاسرسي زېرمه (RAM) د څه شي مخفف دی؟

۳. د سم لاسی لاسرسي زېرمې (RAM) درې مختلف اقسام لیست کړئ؟

۴. د حافظو د لاندې اقسامو څخه کومې حافظې د اصلي حافظې په حیث استعمالېږي؟

a. ROM and SDRAM
b. SRAM and DRAM
c. SDRAM and DRAM
d. DRAM and EPROM

5. ------- د تل له پاره معلومات زېرمه کوي حتی که برق بند هم وی.

a. ROM
b. DRAM
c. RAM
d. SRAM

٦. یواځې لوستونکې زېرمه (ROM) د څه مخفف دی؟

۷. ایا فلش د یوازې لوستونکې زېرمې (ROM) یا سم لاسي لاسرسي زېرمې (RAM) یوه برخه ده؟

۸. د ای ای پی یوازې لوستونکې زېرمه (EEPROM) او ای پي روم (EPROM) تر منځ فرق په څه کي دي؟

۹. یوه حافظه چې 10 Address Lines لري او د هري حافظې په هر ځای باندې 1byte معلومات زیرمه کولی شي نو ددې حافظې ظرفیت به څومره وي؟

۱۰. د ایس سم لاسی لاسرسي زېرمه (SRAM) ابتدايي کاریال (Primary Applications) کوم دي؟

۱۱. د ډی رام (DRAM) ابتدايي کاریال (Primary Applications) کوم یو دی؟

۱۲. لاندېنی اصطلاحات تعریف کړئ؟

a. Track
b. Sector
c. Cluster

۱۳. په یو هارډ ډیسک کې څلور سطحې دي هره سطحه 80 ټریکونه لري او هر ټریک بیا 32 سیکټرونه لري هر سیکټر 512B زېرمه کوي نو ددی ډیسک ظرفیت څومره دی؟

۱۴. د FAT یا File Allocation Table وظیفه څه ده؟

۱۵. په کمپیوټر کې د چټک څخه ورو طرف ته د حافظو اقسام لیست کړئ؟

۱۶. د کېشی حافظې اقسام واضح کړئ؟

۱۷. کوم ډول حافظې د کېشی حافظې له پاره استعمالېږي؟

۱۸. خیالې حافظه واضح کړئ؟

۱۹. د خیالې ادرس او فزیکې ادرس تر منځ فرق ووایئ.؟

۲۰. فزیکې ادرس مشخصوی اندازه د

a. Virtual memory
b. Physical memory
c. Cache memory

۲۱. د خیالي ادرس شکل وښایئ؟

۲۲. هیټ نسبت څه ته وایي؟

۲۳. Temporal Locality واضح کړئ؟

۲۴. Spatial Locality څه ته وایي؟

۲۵. د کېشی حافظې بدلون Cache Mapping میتودونه واضح کړئ؟

۲۶. د هغه ادرس شکل وښایاست چې د کېشی حافظې پواسطه د مستقیمی کړنلاري له پاره لیدل شوی وي؟

۲۷. د کېشی حافظې کړنلاري میتودونه لیست کړئ؟

۲۸. د هغی ادرس شکل وښایاست چې د کېشی حافظې پواسطه د پیوسته کړنلاري سیټ بندي له پاره لیدل شوی وي؟

۲۹. که چیري هره صفحه 8KB ونیسي نو صفحی افسیت یي څومره بیټونه دي؟

۳۰. د صفحو جدول وظیفه څه ده؟

۳۱. په TLB کې کوم معلومات زېرمه کیږي او TLB چېرته زېرمه دی؟

۳۲. د کېشی حافظې کړنلاری میتودونه لیست کړئ؟

۳۳. د کېشـی حافظې د مسـتقیمی کړنلاري په مقابل کې د پیوسـته کړنلاري سیټ بندي ګټې کومې دي؟

۳۴. درې د لیکلو اصـول (Write Policies) چې د حافظې له پاره استعمالیږي واضح کړئ؟

۳۵. ______ د چټکو حافظو یو ډول دی.

a. Cache memory
b. Main memory
c. Secondary memory
d. Hard disk

ستونزې

۱. لاندي اصـلي او کيشی حافظي درکړل شوي، CPU د $0 \times 0, 0 \times 2, 0 \times 3, 0 \times 4,$ $0 \times 5, 0 \times 3, 0 \times 6, 0 \times 6, 0 \times 7, 0 \times B, 0 \times D$ او 0*F ادرسـونه جوړوي نو د کېشی حافظې محتويات وښايي او هيټ نسبت يي پيدا کړئ؟

ADDRESS	Contents
0000	5
0001	0
0010	1
0011	11
0100	15
0101	09
0110	16
0111	23
1000	65
1001	01
1010	8
1011	9
1100	15
1101	0
1110	2
1111	5

v	Tag	Data

۲. لاندي کېشی (Cache) او اصلي (Main Memory) حافظې درکړل شوی: د CPU $0 \times 1, 0 \times 2, 0 \times 1, 0 \times 8, 0 \times 9, 0 \times 1C, 0 \times 1D, 0 \times 3$ او 0*4 ادرسونه جوړوي:

a) د Two-Way Set Associative Mapping په استعمال سره د کيشی (Cache) محتويات وښايي او د LRU پالېسی په نظر کې ونيسئ

b) Hit rate څه شی دی؟

Set Address	V	Tag	B1	B0	LRU	V	Tag	B1	B0	LRU
00	0				0	0				0
01	0				0	0				0
10	0				0	0				0
11	0				0	0				0

Address	Content	Address	Content
00000	5	10000	5
00001	3	10001	0
00010	11	10010	1
00011	6	10011	11
00100	7	10100	15
00101	8	10101	09
00110	9	10110	12
00111	12	10111	23
01000	0	11000	65
01001	0	11001	21
01010	8	11010	8
01011	7	11011	7
01100	9	11100	9
01101	0	11101	0
01110	2	11110	2
01111	5	11111	5

۳. یو کمپیوټر 24 بیټه فزیکي ادرس لري (24bit Physical) ادرس لري او د حافظې هر موقعیت (Memory Location) یې 1byte ادرس نیسي دا کمپیوټر 24bit Cache Lines لري او هر Line یې 16B نیسي د Tag, Index او Byte Offset ادرسونو بڼه یا شکل یې وښایي په استعمال د:

(a) Direct mapping
(b) 4-Way Set Associative
(c) 8-Way Set Associative

۴. یو کمپیوټر 32KB خیالي حافظه لري او 8KB اصلي حافظه لري او د صفحو اندازه یې 512B ده:

a) په خیالي حافظه کې څومره بیټونه دي؟

b) په خیالي حافظه کې څومره صفحې دي؟
c) د فزیکي ادرس له پاره څومره بیټونه ضروري دي؟
d) په اصلي حافظه کې څومره بلاکونه دي؟

۵. یو کمپیوټر 256MB خیالي حافظه , 4MB اصلي حافظه او 8K کېشی حافظه لري او صفحې اندازه یې 2KB په نظر کې ونیسئ:
a) د خیالي ادرس اندازه یې څومره ده؟
b) د فزیکي ادرس اندازه څومره ده؟
c) په خیالي حافظه کې څومره صفحې دي؟
d) په اصلي حافظه کې څومره بلاکونه دي؟
e) د صفحې جدول اندازه څومره ده؟ (البته د ځایونو د تعداد په شمول او دهر ځای مجموعي اندازه سره د ټولو معلوماتو)

۶. یو کمپیوټر 20bits خیالي حافظه لري او هره صفحه یې 2KB ده نو:
a) د خیالي حافظه اندازه یې څومره ده؟
b) په خیالي حافظه کې څومره صفحې ده؟

۷. یو کمپیوټر په اصلي حافظه کې 1M بلاکونه لري او ۴ بلاکونه (4Blocks) لري او په هر بلاک کې ۴ کلمات (4words) وي نو:
a) د فزیکي ادرس اندازه یې څومره ده؟
b) د مستقیمې کړنلاري په استعمال سره د فزیکي ادرس د Tag, Index او Word Offset اندازه مشخص کړئ؟
c) د Two-Way Set Associative Mapping په استعمال سره د یو فزیکي ادرس د Tag او Word Offset اندازه څرګنده کړئ؟

د پروسیس مرکزي واحد (CPU) په ۲٦.۷ شکل کې 00_0 او 0b_0 ادرس جوړوي: فرض کړی 0 صفحه تطبیق کیږي په بلاک 1 کې او صفحه 2 تطبیق کیږي به بلاک 0 کې: د صفحو د جدول محتویات وښایاست.

اتم څپرکی

اسمبلي ژبه او ارم ARM لارښوونې لمړی برخه
Assembly Language and ARM Instructions
(Part I)

موخې: د دې څپرکي په لوستلو سره به وکولای شئ چې:

- د کمپایلر (Compiler) او اسمبلر (Assembler) دندې تشریح کړئ.
- په لوړ لېول ژبو (HLL[1]) کې لیکل شوی کود د ماشین ژبې (Machine Language) ته تبدیل کړئ.
- د ARM پروسیسر جوړښت څرګند کړای شئ.
- د پروسیسر د حالت راجیسټر (PSR[2]) دنده تشریح کړئ.
- د عمل ورباندې تر سره کېدونکو (Operands) له نظره د لارښوونو ډولونه وپېژنئ.
- د ARM د لارښوونو له بېلا بېلو ډولونو سره آشنا شئ.

---

[1] **HLL: High Level Language**
[2] **PSR: Processor State Register**

- د شرطیه لارښوونو کړنې تشریح کړای شئ.
- پـه لوړ لېول ژبو (HLL[3]) کې لیکـل شـوې کود د اسـمبلي ژبې(Assembly Language) ته تبدیل کړای شئ.
- د شیفټ(Shift) او روټیټ(Rotate) لارښوونې تشریح کړای شئ.
- د سټیک (Stack) د لارښوونو کړنې تشریح کړای شئ.
- د برانچ(Branch) د لارښوونو اپلیکیشنونه تشریح کړئ.

۸,۱ پېژندګلوي

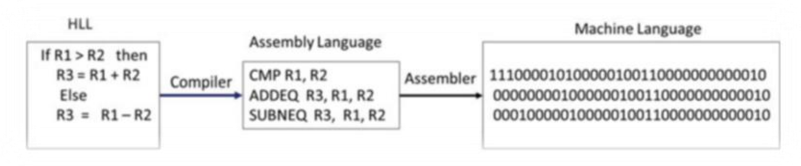
پروګرام لیکونکي او پروګرام جوړونکي د پروګرامونو او پوسـتغـالو د جوړولو پـه لـپـاره پـه لـوړه کـچـه د پـروګـرام لـیـکـنې ژبې (High Level Programming Languages) استفاده کوي، مګر د دې لپاره چې یو پروګرام د ماشین په واسطه د اجرا کېدلو وړ شي نوموړی پروګرام باید په ماشین ژبه (Binary Language) کې وي.
د ۱.۸ شـکل په یوه لوړه کچه د پروګرامېنګ ژبه کې لیکل شـوی کود د ژباړې شـکل ښـکارندویی کوي چې ښـکته کچې ته ژباړه شوی دی. یو کمپایلر په لوړه سطحه ژبو کې لیکل شـوی کود د اسـمبلي ژبې ته ترجمه کوي چې وروسـته نوموړی کود د اسـمبلر په مرسته د ماشین ژبې (0,1) ته ترجمه کېږي.
هـرمـرکـزي پـروسـیـس کوونکي واحـد (CPU) د لارښـوونو یوه ټولګـه (Set Of Instructions) لري چې لـه مخې یې د نوموړي مرکزي پروسـیس کوونکې واحـد (CPU) د کړنو او عملیو ډول ښـکاره کېږي. د مرکزي پروسیس کوونکي واحد (CPU) نوموړې لارښـوونې په Mnemonicډول یا هم د ځانګړو مخففاتو په ډول سـره ښـکاره کوي د بېلګې په توګه د جمعې لپاره د ADD او د منفي لپاره د SUB له مخففاتو څخه استفاده کېږي.

د ADD R1, R2, R3 لارښوونه په داسې ډول سره تعبیر کېږي چې د $R2$ او $R3$

---

[3] **HLL: High Level Language**

مجموعه په $R1$ کې باید وساتل شي، په دې ځای کې $R3$ او $R1, R2$ د عمل ورباندې ترسره کېدونکې په نوم سره یادېږي.



۱،۸ انځور د HLL تبدیلول دوه په قاعدی ژبې ته

لاندې لیکل شوي HLL اړوندې اسمبلي ژبې ته ژباړه شوي دي.

په لوړه سطحه لیکل شوی کود HLL	اړونده د اسمبلي ژبې کود
ADD R3, R1, R2	R3 = R2 + R1
SUB R3, R1, R2	R3 = R1- R2

پروګرام لیکونکي د اسـمبلي ژبې د لیکلو لپـاره له لاربـوونو څخه اسـتفاده کوي. او د نوموړې ژبې مشهور دندی په لاندې ډول سره دي.

- د لوړترین سرعت لرونکي کوډ د لیکلو لپاره له اسمبلي ژبې څخه استفاده کېږي.
- د لوړې کچې ژبو (HLL) په ښه پوهېدلو کې له پروګرام لیکونکي سره مرسته کوي.
- د لوړې کچې ژبو (tt) لپاره د کمپایلر جوړول او منځ ته راوړل د اسمبلي ژبی ته اړتیا لري.
- په ډیری سیستمونو کې د ډرایورونو د جوړولو لپاره له همدې ژبې څخه کار اخیستل کېږي.
- په داسې حالاتو کې چې د لوړې کچې ژبې (HLL) سختغالي (Hardware) ته د لاسرسي اجازه ورنه کړي له اسمبلي ژبی څخه کار اخیستل کېږي.

۲,۸ د لارښوونو د سېټ جوړښت (ISA)

د مرکزي پروســیس کوونکي واحـد (CPU) تولید کوونکی کمپنۍ له مرکزي پروسـیس کوونکي واحد(CPU) سـره یوه لیکنه هم چاپوي چې په هغې کې د نوموړې مرکزي پروســیس کوونکي واحـد (CPU) اړونـد ځینې معلومات لکه دراجســترونو لېســت، د هر راجســتر دنده، د ډېټا وصــلونکي اندازه، د آدرس وصــلونکي اندازه او د هغه لارښــوونو لېســت چې د نوموړې مرکزي پروسـیس کوونکي واحد (CPU) په واسـطه د ترسـره کېدلو وړ وي، شـتون لري. هر مرکزي پروسـیس کوونکي واحد (CPU) یو پېژندل شوی د لارښـوونو د سـېټ جوړښت (ISA) لري چې له همدې څخه په استفادې یو پروګرام لیکونکی شــخص کولای شــي نوموړې مرکزی پروســیس کوونکی واحد (CPU) ته د اســمبلي ژبې کود لیکنه تر ســره کړي. د هر پروســیســر د لارښــوونو ســېټ د همغه پروســیســر لپاره ځانګړی وي. د بېلګې په توګه Pentium پروسیسر نسبت ARM پروسیسر ته متفاوت د لارښوونو سېټ لري. د نوموړو لارښــوونو طبقه بندي د دوه لاملونو په اســاس تر ســره کېږي چې لمړی د عمـل وربانـدې ترســره کېـدونکو تعداد دی او دوهم د عملیې (Operations) ډول دی.

۱,۲,۸ د لارښــوونو طبقـه بنـدي د عمل وربانـدی ترســره کیدونکو د تعداد له نظره (Classification of Instruction Based on Number of Operands): بغیر

له کوم عمل وربانـدی ترســره کیدونکی(Operand)څخه لارښــوونې: لاندې ځینې لارښــوونې دي چې کوم عمل وربانـدې ترسـره کېدونکی (Operand) ته اړتیا نه لری.

لارښوونه (Instruction)	اړونده کړنه (Operation)
HALT	Halt The CPU
NOP	هیڅ کومه کړنه نه ترسره کوي
PUSH	د یو څیز اضافه کول Stack ته
POP	د یو څیز کمول د Stack له سر څخه

د یو عمل ورباندی ترسره کیدونکي (Operand) لرونکې لارښوونې: په لاندې توګه ځینې لارښوونې دي چې یوازې یو عمل عمل ورباندی ترسره کیدونکی Operand ته اړتیا لري.

INC	Operand	Example: INC R1 – Increment register R1 by 1
DEC	Operand	Example: DEC R1 – Decrement register R1 by 1
J	Operand	Jump to memory location labeled by target
ADD	Operand	Add operand to the accumulator (ACC) ACC → ACC + operand

د دوه عمل وربانديې ترسره کیدونکي (Operands) لارښوونې: په لاندې توګه ځینې لارښوونې دي چې دوه عمل ورباندی ترسره کیدونکو Operands ته اړتیا لري یا هم ویلای شو چې په دوه عمل ورباندی ترسره کوونکو Operandsباندې عملیې اجرا کوي.

لار ښوونه (Instruction)	اړونده عملیه (Operation)
ADD R1, R2	R! = R1 + R2
MOV EAX, EBX	EAX ← EBX

د یادونې وړ ده چې اینټل ($Intel$) جوړښت دوه عمل ورباندی ترسره کیدونکو $Operands$ لرونکي لارښوونې استفاده کوي.

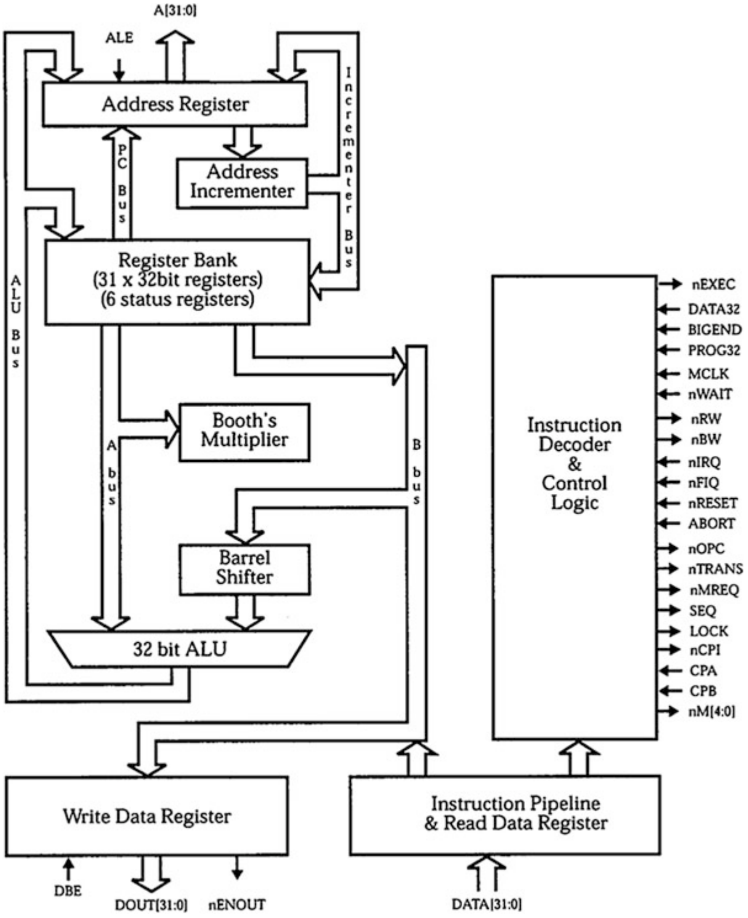
د دریو عمل وربانديِ ترسره کیدونکو $Operands$ لارښوونې: ډیری عصري پروسیسرونه لکه: $ARM, MIPS$ او $Itanium$ د دریو عمل وربانديِ ترسره کیدونکو Operands لرونکو لارښوونو څخه استفاده کوي. د بېلګې په توګه: ADD R1, $R2$, R3 د نوموړې لارښوونې څخه موخه دا ده چې د R2 + R3 نتیجه په $R1$ رېجیسټر کې باید وساتل شي.

۳,۸ د ارم پروسیسر جوړښت (ARM Processor Architecture)

د Advanced RISC Machine (ARM) پروسیسر د لومړي ځل لپاره د Acorn کمپنۍ په واسطه منځ ته راغلو چې نن سبا په نړیواله سطحه د ډیروکوچني پرسیسرونو (Microprocessor) د جوړښت اساسي برخه ګڼل کېږي. ARMد یوه پروسیسر هسته منځ ته راوړي په زرګونو تولیدونکو کمپنیو نوموړې هستې ته نورې ډېرې بېلا بېلې مهمې برخې اضافه کړې دي. په عمومي ډول سره ARMدوه ډوله لارښوونې استعمالوي چې یو ته یې Thumb او بل ته یې $Thumb$-2 وايي. Thumb لارښوونې 16-bit او Thumb-2 لارښوونې بیا 32-bit دي. د یادونې وړ ده چې نن سبا ډیری پروسیسرونه د ARM له Thumb-2 جوړښت یعنې 32-bit څخه استفاده کوي. ARM پروسیسر په ټوله کې ۱٦ عدده راجسټرونه لري چې له R0-$R15$ پورې په Rn نومونو سره نومول شوي دي او له $R0$-$R12$ پورې راجسټرونو ته د عامه موخو لپاره استفاده کېدونکي پروسیسرونه وايي. ARM د دې وړتیا لري چې Thumb لارښوونې تر سره کړي چې د نوموړو لارښوونو د ترسره کولو پر وخت له $R0$-$R7$ پورې راجسټرونو څخه استفاده کوي. او همدانګه کولای شي چې د Thumb-2 لارښوونې اجرا کړی.

ARM د داسې آلو او آپلیکیشنونو لپاره استعمالیږی چې قوي پروسیسرونو ته اړتیا لري لکه:

ټیلیفوني اړیکې، د معلوماتو اړیکې، د لېږد وړ کمپیوټرونه او څیرک کارتونه. په ساده ډول سره ویلای شو چې ARM اصلا یو 32-bit RISC پروسیسر دی.



۲,۸ انځور د ARM7 پروسیسر د جوړښت بلاک دیاګرام

یو 32-bit ډېټا وصـلونکی او آدرس وصـلونکی حقیقي وخت کې ځواب ورکونکو (Real Time) اپلیکیشنونو لپاره استفاده کېږي.

د لارښـــوونو اصـــلی بنې ته تبدیلونکی او د منطق اداره کوونکی (Instruction Decoder And Logic Control): د نوموړې برخې دنده د لارښـوونو اصلي بڼې تـه تبدیلول او د پروســیســـر نورو برخو ته د کړنو د اجرا کولو په موخه د کنټرول او اداره کوونکو سیګنالونو استول دي.

آدرس راجستړ (Address Register): د آدرس وصلونکي لپاره یو $32\text{-}bit$ آدرس ساتلو دنده پر غاړه لري.

د آدرس زیـاتونکی (Address Increment): د دې لپـاره اســتفـاده کېږي چې یو آدرس د څلورو واحدونو په اندازه زیات او په آدرس راجستړ کې یې ذخیره کړي.

د راجســتړونو بانک (Register Bank): نوموړې برخه 31 بېلا بېل 32-bit راجستړونه لري چې د دې له جملې څخه 6دانې یې د حالت ښکاره کوونکي (Status) راجستړونه دي.

د بیرل اړولو برخه (Barrel Shifter): نوموړې برخه د لوړ سرعت ادلون بدلون کولو (ټیله کولو) لپاره استعمالیږي.

د حســـابي او منطقی برخه (ALU): یوه 32-bit برخه د حســـابي او منطقي عملیو د تر سره کولو لپاره استعمالیږي او د کمپیوټر د پروسیسر تر ټولو مهمه برخه ګڼل کېږي.

د معلوماتو د لیکلو راجسـتړ (Write Data Register): کله چې پروسـیسر کوم بل ځای ته د معلوماتو د لیږدولو کړنه تر سـره کوي نوموړې معلومات په همدې راجسـتړ کې ځای پر ځای کوي.

د معلوماتو د لوســتلو راجسـتړ (Read Data Register): کله چې پروسـیســر د یو ځـای څخـه د بېلګې په توګه له حافظې څخه د معلوماتو د لوســتلو عملیه تر ســره کوي نوموړي معلومات همدا راجستړ ذخیره کوي.

د ARM د عملیو د اجر کولو حالات:

د استعمالونکي طریقه (User Mode): د نارمل او عادي کړنو د تر سره کولو لپاره نوموړي طریقې څخه استفاده کیږي.

د IRQ حالت: د مزاحمت Interrupts د عملیو د اداره کولو لپاره نوموړي حالت ډیزاین شوی دې.

د Supervisory موډ: نوموړي موډ د عامل سیستم (Operating system) په واسطه استفاده کیږي.

د FIQ حالت: نوموړی موډ د لوړ سرعت لرونو مزاحمتونو Interruptsلپاره استفاده کیږي.

د نا تعریف شوې طریقه (Undefined Mode): نوموړی طریقه په داسې حالاتو کې استفاده کیږي چې یوه نامعلومه لارښوونه اجرا کیږی.

د (Abort Mode) طریقه: کله چې پروسیسر د داسې معلوماتو غوښتنه وکړي چې لا حافظې(Memory) ته نه وي رارسیدلي او پروسیسر د هغی معلوماتو د انتقال لپاره چی حافظې (Memory) ته یي انتقالوي لا زیات وخت ته اړتیا لري، په داسې حالاتو کې له نوموړي طریقې څخه استفاده کیږي.

۴,۸ د ارم راجسټرونه (ARM Registers)

ARM7 ټول ۳۱ عمومي راجسترونه لری او ٦ موقعیتي راجسترونه لري. د استعمالونکي لپاره ۱٦ عمومي او یو موقت راجستر (PSR) د پروګرامر لیکلو لپاره موجود دی. دا راجسترونه د R0 نه تر R15 پوری نامګذاري شوي. R15. د پروګرام شمیرونکي (PC) لپاره استعمالیږی، R14 د راجسترونو د وصلولو لپاره استعمالیږي، او R13 د

ستیک نښه کونکي (SP) لپاره استعمالیږی. ۳٫۸ شکل د استعمالونکی طریقی روښانوي. د اوسـني پروګرام د حالت ښـکاره کوونکی راجستر (CPSR). د ۴٫۸ شکل د پروګرام د حالت د ښـکاره کوونکي راجستر جوړښـت روښـانه کوي. نوموړی راجستر د کنټرول او فلګ بیټونو د ســاتلو لپاره اســتفاده کیږي. د فلګ بېټونه عبارت دي له N, Z, C او V څخـه او اداره کوونکی بیټونـه یي عبـارت دي له I, F او M0-M4. د یادونې وړ ده چې فلګ بېټونه د حسابي، منطقي او یا هم مقایسوي عملیو په پایله کې تغیر کوي.

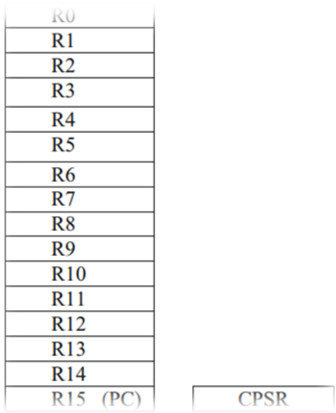
د فلګ بېټونه (Flag Bits):

د N *بېـټ:* د نوموړي بېـټ هـدف منفي $(Negative)$ دی، کـه چېرتـه $N = 1$ وي مطلب دا چې د یوې عملیې پایله منفي ده او که چیرته $N = 0$ وي مطلب دا چې د یوې عملیې پایله مثبت ده.

د Z *بېټ:* د نوموړي بېټ هدف صــفر $(Zero)$ دی، که چیرته $Z = 1$ وي مطلب دا چې د یوې عملیې پایله صـفر ده او که چیرته $Z = 0$ وي مطلب دا چې د یوې عملیې پایله صفر نه ده.

د C *بېټ:* د نوموړې بېټ هدف حاصل $(Carry)$ ده، که چېرتـه $C = 1$ وي مطلب دا چې د یوې عملیې پایله یو حاصل لري او که چیرته $C = 0$ وي مطلب دا چې د یوې عملیې پایله حاصل نه لري.

د V *بېټ:* د نوموړي بېټ هدف زیاتبدل $(Overflow)$ دي، که چیرته $V = 1$ وي مطلب دا چې د یوې عملیې پایلې Overflow کړی او که چیرته $V = 0$ وي مطلب دا چې overflow یې نده کړی.



۳,۸ انځور: د استفاده کوونکي موډ راجسټرونه

31	30	29	28	27	7	6	5	4	3	2	1	0
N	Z	C	V	Unused	I	F	T	M4	M3	M2	M1	M0

۴,۸ انځور: د $CPSR$ جوړښت

اداره کوونکی بېټونه (Control Bits):

د I *بېټ:* د دي بېټ هدف د مزاحمت (Interrupt) بېټ دی او کله چې نوموړی بېټ د 1 قېمت وساتي مطلب دا چې پروسیسر د کوم پوستغالي (Software) څخه مزاحمت (Interrupt) نه شي تر لاسه کولای.

د F *بېټ:* د FIQ یعنې لوړ سرعت لرونکي مزاحمت ($Interrupt$) د غوښتلو موډ د فعالولو او غیر فعالولو لپاره استفاده کیږي.
M0, M1, M2, M3, M4 بېټونه د موډ د بېټونو په نومونه سره یادیږي د او استفاده کوونکي د موډ (Mode User) لپاره له 10000 سره مساوي دي.

د T *بېټ:* نوموړی بېټ ته د حالت (State) بېټ هم وايي او کله چې T = 1 سره مطلب دا چې په پروسیسر کې د $Thumb$ لارښوونې دي او کله چې T = 0 شي مطلب دا چې په پروسیسر کې د ARM لارښوونې دي.

۵,۸ د آرم لارښوونې (ARM Instructions)

د آرم جوړښت دوه ډوله د لارښوونو سېټ Thumb 16-bit او Thumb 32-bit قبلوي. د آرم ډېرې لارښوونې عمل ورباندې ترسره کېدونکي (Operands) باندې کړنې تر سره کوي. نوموړې لارښوونې د لارښوونو د جوړښت او دندو له مخې په لاندې لويو گروپونو باندې ويشل شوې دي:

A) د ډېټا د پروسیس کولو لارښوونې (Data Processing Instructions)
B) د يوازې معلوماتو تبادله (Single Data Swap)
C) د شيفټ(Shift)او روټيټ (Rotate) لارښوونې
D) شـرطي او غير شـرطي لارښـوونې (Conditional And Unconditional Instructions)
E) د سټېک عمليې (Stack Operations)
F) Branch
G) ضربي لارښوونې (Multiply Instructions)
H) د ډېټا د لېږد رالېږد لارښوونې (Data Transfer Instructions)

۱,۵,۸ د ډېټا د پروسیس کولو لارښوونې (Data Processing Instructions):

د ډېټا د پروسيس کولو لارښوونې عبارت دي له: $AND, EOR, SUB, RSB, ADD, ADC, SBC, RSB, TST, TEQ$ $CMP, CM, N\ MOV, BIC, MNW$ او MNW څخه. نوموړې لارښوونې د راجستر Operands او Immediate Operands څخه استفاده

کوي او عمومي شکل يې په لاندې ډول سره دی:

Mnemonic {S} {شرط} Rd, Rn, operand2

نمانیک (Mnemonic): د یوې کړنې لپاره یو مخفف دی د بېلګې په توګه: ADD د جمعې د عملیې د تر سره کولو لپاره او SUB د منفي عملیې د ترسره کولو سمبولونه دي.

{ }: د دې دوه قوسونو تر منځ د کمانډونو شتون اختیاري دی د بیلګې په توګه په پورته عمومي شکل کې **S** او شرط.

کله چې په لاربښوونو کې **S** شتون ولري مطلب يې دا دې چې د پروسیسر د حالت بېټ (PSR) قیمت باید نوی شي.

Rd: له دې څخه هدف مطلوب (Destination) راجستر دی.

Rn: له دې څخه هدف لومړی عمل وربانديې ترسره کیدونکی **(**Operand) دی.

دوهم عمل وربانډی ترسره کیدونکی (**2** Operand**)** کېدلای شي چې په کوم یوه بل راجستر کې شتون ولري او یا هم تصادفي قیمت و لري.

الف: د راجسترونو عمل وربانډی ترسره کیدونکی (Register Operands):

عملیه همیش په راجستر کی ترسره کیږی. لومړی راجستر مطلوب (Destination) راجستر دی دوهم راجستر د لمړی عمل وربانډی ترسره کیدونکی (Operand1) او دریم راجستر د دوهم عمل وربانډی ترسره کیدونکی (Operand2) لپاره دی.

لاندی د راجستر د عمل وربانډی ترسره کیدونکي په بڼه حسابي او منطقي عملیې دي.

$ADD\ R0, R1, R2; R0 = R1 + R2$ یعنې د لمړي راجستر ($R1$) قیمت او دوهم راجستر ($R2$) قیمت سره جمع کړه او په دریم راجستر (R0) کی یی زخیره کړه

$ADC\ R0, R1, R2; R0 = R1 + R2 + C$ یعنې د $R1$ او $R2$ راجسترونو قیمتونو سره له حاصل یوځای کړه

$SUB\ R0, R2, R3, R2; R0 = R2 - R3$ چیرته چې R2 لمړی عمل وربانډی

ترسره کیدونکی او R3 دوهم عمل وربا‌ندی ترسره کیدونکی دی

$SBC\ R0, R2, R3, R2; R0 = R2 - R3 + C - 1$ تفریق سره له حاصل.

$RSB\ R0, R2, R3, R2\ \ ;\ \ R0 = R5 - R2$ د تفریق معکوسه عملیه.

$RSC\ R0, R2, R5\ \ ;\ \ R0 = R5 - R2 + C - 1$ د تفریق معکوســه عملیه د حاصل سره.

AND R0, R3, R5; R0= R3 AND R5.
ORR R7, R3, R5; R7=R3 OR R5.
EOR R0, R1, R2; R0 = R1 Exclusive OR with R2.

BIC R0, R1, R2 بیټ پاکونکی کوم چی دوهم عمل وربا‌ندی ترســره کیدونکی کی مربوطه بیـټ پـه لومړی عمـل وربـاندی ترســره کیدونکي کی پاکوي او نتیجه یی په مطلوب راجستر کی زیرمه کوي.

۸٫۱ مثال: فرض کړی چې R1 محتوا 1111111111011111 او د R2 محتوا 0011 1110 0100 1000 دی نو د BIC له ترسره کیدو څخه وروسته R0, R1, R2 د R0 Contains 0111 101100011100.

الف: سمدستي قیمت (Immediate Value)

په ســمدســتي(Immediate) عمل وربا‌ندی ترســره کیدونکی کی دوهم عمل وربا‌ندی ترسره کیدونکی یو سمدستي(Immediate) قیمت لری چې تر ۱۲ بیټونو پوری کیدلی شی.

$ADD\ R1, R2, \#\&25;$
$R1 - R2 + \& amp; amp; amp; amp; amp; 25, \#means\ immediate\ and\ \& amp; amp; amp; amp; amp\ means\ the\ immediate\ value\ is\ hexadecimal.$
$AND\ R2, R3, \#\&45;\ \ R2 = R3\ AND\ \&45.$
$EOR\ R2, R3, \#\&45;\ \ R2 = R3\ Exclusive\ OR\ \&45.$

۲،۸ مثال: فرض کړئ چې R2 محتوا 0x12345678 دی د لاندی لارښوونی د ترسره کیدو څخه وروسته به د R1 قیمت څه وي؟

ADD R1, R2, #0x345

د ADD لارښوونه به د R2 سره 0x345 جمع کوي او نتیجه یی په R1 کې زخیره کوي.

د PSR د فلګ بیټ ترتیبات (Setting Flag Bits of PSR) پورتني لارښوونه د PSR فلګ بیټ باندی کوم تاثیر نه لري څکه چې دغه لارښوونه د S کوم انتخاب نه لري د S وروستاړي په اضافه کولو سره به دغه لارښوونه د PSR فلګ بیټ کی تغیر راولي.

$ADDS\ R1, R2, R3;\ The\ suffix\ S\ means\ set\ appropriate\ flag\ bit.$
$SUBS\ R1, R2, R3;\ This\ will\ set\ zero\ flag\ to\ 1.$

۲،۵،۸ مقایسوی او ازمایښتی لارښوونی (Compare And Test Instruction): د آرم پروسیسر مقایسوی او ازمایښتي لارښوونې ددې لپاره استعمالوي چې د PSR فلګ بیټ (bits) تعین کړي چی په لاندی ډول دی:

CMP, CMN, TST او TEQ.

دغه لارښوونې دوه عمل ورباندی ترسره کیدونکی د مقایسی او ازمایښت لپاره استعمالوي چې د دوی نتیجه په بل کوم راجستر کی نه ثبتیږي

CMP لارښوونه (مقایسوی لارښوونه) د CMP لارښوونی شکل په لاندی ډول دی:

CMP Operand1, Operand2: د CMP لارښوونه لمړی عمل وربانډی ترسره کیدونکی (operand1) د دوهم عمل وربانډی ترسره کیدونکی (Operand2) سره مقایسه کوي دغه لارښوونه دوهم عمل وربانډی ترسره کیدونکی (Opernad2) له لمړی عمل وربانډی ترسره کیدونکی (Operand1) څخه تفریق کوي او مناسب فلګ ورته تعین کیږي د فلګ بیټ ددې عملیی په نتیجه باندی تعین کیږی داسې چې Z فلګ هغه وخت ورته تعین کیږی کله چې د لمړي عمل کوونکي قیمت د دوهم عمل کوونکي

قیمت سره مساوي وي.

N فلګ هغه وخت ورته تعین کیږی کله چې د لمړی عمل کوونکی قیمت د دوهم عمل کوونکی لـه قیمـت څخه کوچنې وی C فلګ هغه وخت ورته تعین کیږي کله چې د دی عملیی په نتیجه کې حاصل راشي.

۴,۸ مثال: فرض کړئ چې $R1\ contains\ 0x00000024$ او $R2\ conatains\ 0x00000078$ د $CMP\ \ R1, R2$ عملیه به د **N** فلګ به **1** ته تعین کړي

CMP Rd, immediate value سمدستي (Immediate) قیمت کیدلای شي چې ۸ بیټه وي لکه:

CMP R1, #0xFF

CMN Compare Negate د CMN شکل په لاندې ډول دی:

CMN Operand1, Operand2

دغه عملیه به لمړی عمل ورباندې ترسره کیدونکي (operand1) او دوهم عمل ورباندې ترسره کیدونکي (operand2) سره جمع کوي او مناسب فلګ ورته تعینوي

۳,۵,۸ مثال: فرض کړئ چې $R1\ contains\ 0x00000024$ او $R2\ contains\ 0x13458978$ د $CMN\ R1, R2$ نتیجه به حاصل وی چې د Cفلګ به **1** ته تعین کړي

TST (Test Instruction) د ازمایښتی لارښوونې شکل په لاندی ډول دي:

TST Oprand1, Operand2

دغه لارښـــوونه د انډ (AND) عملیه د لمړي عمل ورباندی ترســـره کیدونکي (operand1) او دوهم عمل ورباندی ترسره کیدونکي (operand2) ترمنځ ترسره کوي

او مناسـب فلګ ورته تعينوي عمل وربان‌دی ترسـره کيدونکی (Operand) کيدلی شـي راجسـتر وي سـمدستي تصادفي قيمت اخلي په دي شکل ;TST R1, R2 دغه لارښوونه د R1 AND R2 عمليه ترسره کوي او مناسب فلګ ورته تعينوي يا

Tst R1, Immediate چې Immediate قيمت ۸ بيت وي داسـی چې ,TST R1 0xFF

TEQ R1, R2 دغه لارښوونه د ايکسکلوسيو آر علميه ترسره کوي.

که چيرته R1 قيمت له R2 قيمت سره مساوي وي نو د Z فلګ **1** ته تعينيږي.

۳,۵,۸ د راجسـترونو تبادله کولو لارښـوونې (Register Swap Instruction (MOV,MVN)): د راجسترونو تبادله کولو لارښوونی په عمومی ډول لاندی شکل لري:

الف: MOV{S}{Condition} Rd, Rm

د محتوی (Contents) ليږل له Rm څخه Rd ته.

۶,۸ مثال: د R1 متحوی (Content) به د لاندې لارښوونې د ترسره کيدو څخه وروسته به څه شی وي؟

فرض کړئ R2 Contains 0X0000FFFF

الف: R2→MOV R1, R2 ; R1

R2 = 0x0000FFFF

ب: NOT R2 →MVN R1, R2 ; R1

R2= 0xFFFF0000

$\boldsymbol{A}.\ MOV\ \{S\}\ \{codition\}\ Rd, immediate\ value$
$immediate\ value\ is\ 16\ bits, the\ range\ of\ immediate\ value$
$if\ from\ 0x00000000\ to\ 0x0000FFFF.$

۷,۸ مثال:

$MOV\ R2, \#0x45, the\ contents\ of\ R2\ will\ be\ 0x00000045$

$\boldsymbol{B}.\ MOV\ Rn, Rm, lsl\ \#n\ ; Shift\ Rm\ n\ times\ to\ the\ left$
$and\ store\ the\ result\ Rn.$

$\boldsymbol{C}.\ MOVEQ\ R2, 0x56\ ; if\ zero\ bit\ is\ set$
$then\ executes\ MOVEQ$

۴,۵,۸ شـیفټ او روټیټ لارښووني (Shift And Rotate Instruction): آرم د شـیفټ (Shift) او روټیټ (Rotate) عملیی له نورو لارښـوونو سـره یوځای کړی دی د آرم پروسیسر لاندی د شیفټ (Shift) عملیی ترسره کوي:

LSL	Logical Shift Left
LSR	Logical Shift Right
ASR	Arithmetic Shift Right
ROR	Rotate Right

منطقی شـیفټ لیفټ (Logical Shift left): څرنګه چې په ۵,۸ شـکل کې ښوودل شوي په دي عملیه کې د راجستر هر بیټ چپ طرف ته لیږدول شوی او صفر به لږ تر لږه د پام وړ بیټ کی ځای پر ځای شـي ، منطقی شـیفټ لیفټ د راجسـتر محتوا په دوو کی ضربوي.

$LSL\ \ R1, R1, n, Shift\ \ left\ \ R1\ n\ times\ \ and\ \ store\ \ result$

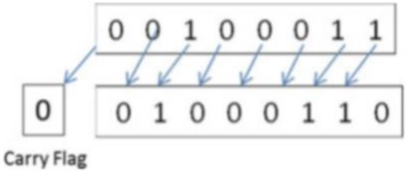
۸,۸ مثال: فرض کړئ چې R1 محتوا x000005000 ده نو د R1 محتوا به وروسـته د لاندی لارښوونې د ترسره کیدو څخه وروسته څه وي؟

LSL R1, R1, 8

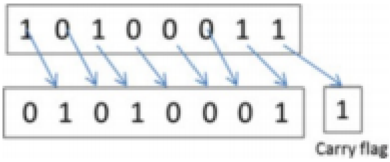
$$R1 = 0x00050000$$

منطقي شـیفټ رایټ ((Logical Shift Right (LSR)): څرنگه چې په ۸,٦ شـکل کی ښودل شوي په دې عملیه کې د راجسټر هر بیټ ښي طرف ته لیږدول شوی او صفر به په خورا مهم بیټ کی ځای پر ځای شــي، منطقی شــیفټ رایټ د راجســټر محتوا په دوو ویشي.

$$LSR\ R1, R1, n, Shift\ to\ right\ R1\ n\ times\ and\ store\ result\ in\ R1$$



۵,۸ انځور منطقي شیفټ لیفټ



٦,۸ انځور منطقی شیفټ رایټ

۹,۸ مثال: فرض کړی چې د R1 محتوا 0x00000500 دی نو د لاندې لارښــوونې د ترسره کیدو څخه وروسته به د R1 محتوا څه وي؟

LSR R1, R1, 4
R1 = 0x00000050

ریاضـیکي شـیفټ رایټ ((Arithmetic Shift Right (ASR)): لکه څرنگه چې په ۷,۸ شـکل کی ښــوودل شــوي دي په ریاضــیکي شــیفټ رایټ کې خورا مهم بیټ نه بدلیږي او هر بیټ ښي طرف ته لیږدول شوی.

روټیټ رایټ (Rotate Right): ۸,۸ شکل ۸ - بیټ راجسټر ښايي او ۸,۹ شکل د یو ځل روټیټینګ(Rotating) څخه وروسته راجسټر ښايي.

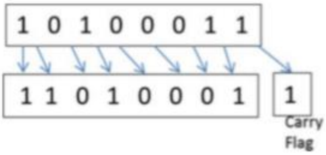
۱۰,۸ مثال: فرض کړئ چې R1 محتوا 0X0000FFFF دی د ۱٦ ځله روټیټینګ (Rotating) څخه وروسته به د R1 محتوا څه وي؟

ROR R1, R1 , #16

R1 = 0xFFFF0000

ارم د معلوماتو د پروسيس کولو لارښووني او شيفټ عملیې سره یو ځای کوي او د شيفټ عملیې د لارښووني په دوهم عمل ورباندې ترسره کیدونکي باندې تطبيق کيږي.

۱۱,۸ مثال: R2 راجسټر محتوا 0XEEEEFFFF دی د لاندی لارښووني ترسره کیدلو به د ۱٦ ځله روټیټینګ (Rotating) کولو سره به د R1 محتوا xFFFFEEE وي.



۷,۸ انځور رياضيکی شيفټ رایټ

۸,۸ انځور روټیټ رایټ عملیي

۹,۸ انځور د یو بیټ روټیټ رایټ عملیي

*ADD R*1, *R*2, *R*3, *LSL* #4 ;   *R*1= R2 þ R3 x 24, R3 is Shifted 4 Times to

The Left And Result Is Added To R3 And Placed In R1

همدارنگه راجستر کولی شي هغه شمیر وساتي چې دوهم عمل کوونکی باید لیږدول شوی وي:

ADD R1, R2, R3, LSL R4	R1¼ R2 þ R3 X 2R4, Number of .times R3 to; be shifted is in R4
MOV R0, R1, LSL #3	Shift R1 to the left three times and .move the; result to R0

۵,۵,۸ د آرم شــرطیـه او غیر شــرطیـه لارښــوونې (ARM Unconditional Instructions And Conditional Instructions): ۱۰,۸ شـــکـل د آرم لارښوونې عمومي شکل ښکاره کوي.

آرم لارښوونی په دوه ډوله دي چې عبارت دي له:

a. غیر شرطیه لارښوونې (Unconditional Instruction)

b. شرطیه لارښوونې (Conditional Instruction)

شـرطیه کود د لارښـوونې ډول مشخصوي که چیرته دغه فیلډ (Field) ۱۱۱۰ وي نو دغه لارښوونه غیر شرطیه لارښوونه ده بغیر له دي لارښوونه شرطیه لارښوونه ده. ددی لپاره چې شــرطیه لارښــوونې څخه کار واخلو نو شــرط به د لارښــوونو وروســتاړی لری چې دغه وروستاړي په لاندې ډول دي:

<u>Condition Code Condition</u>
0000 EQ Equal
0001 NE Not Equal
0010 CS Carry Set
0111 CC Carry Is Clear

0100 MI Negative (N flag is set)
0101 PL Positive (N flag is zero)
0110 VS Overflow Set
0111 VC overflow Is Clear
1000 HI Higher For Unsigned Number
1001 LS Less Than For Unsigned Number
1010 GT Greater For Signed Number
1011 LT Signed Less Than
GT Greater Than ۱۱۰۰
1101 LE less than or equal
1110 AL unconditional instructions
1111 Unused code

Condition Code	Instruction
31 28	27 1

۱۰،۸ انځور د آرم د لارښووني عمومي شکل

مخکې له دې چې لارښـوونه ترسـره شـي لمړی پروسیسر د شرط فلګ ګوري که چیرته د شـرط له لارښووني سره برابر وي نو پروسیسر دغه لارښوونه ترسره کوي او که ورسره برابره نه وي نو پروسیسر دغه لارښوونه نه ترسره کوي

ADDEQ R1, R2, R3; که چیرته د صفر فلګ (Zero flag) تعین شوی وي نو دغه لارښوونه سرته رسیږي.

۱۰،۸ مثال: لاندی په لوړ لیول (HLL) ژبی کی لیکل شــوی کود ارم اســمبلی ژبی ته واړوي:

If R1 = R2 then
ADD R3, R4, R5
 Endif

د آرم اسمبلی ژبې کوډ به د پورتني پروګرام لپاره په لاندی ډول دی:

CMP R1, R2
ADDEQ R3, R4, R5

۱۱,۸ مثال: لاندی په لوړ لیول (HLL) ژبې کې لیکل شـوی کود ارم اسـمبلی ژبې ته واړوئ:

If R1 = R2 Then R3= R4-R5
 Else If R1>R2 Then R3 = R4þR5

د آرم اسمبلي ژبې کوډ به د پورتني پروګرام لپاره په لاندې ډول دی:

CMP R1, R2
SUBEQ R3, R4, R5

ADDGT R3, R4, R5

٦,٨ د آرم د معلوماتو د پروسیس کوونکو لارښوونو جوړښت

د لارښوونو جوړښت د اسمبلر په واسطه ددې لپاره استعمالیږي ترڅو لارښوونې ماشین کود ته واړوي چې ۱۱,۸ شکل د معلوماتو د پروسیس کوونکو لارښوونو جوړښت څرګندوی.

31–28	27 26	25	24–21	20	19–16	15–12	11–0
Cond	0 0	I	Op code	S	Rn	RD	Operand 2

۱۱,۸ انځور د معلوماتو د پروسیس کوونکو لارښوونو شکل

11–7	6–5	4	3–0
# shift	SH	0	Rm

۱۲,۸ انځور د دوهم عمل ورباندی ترسره کیدونکي (Operand2's) کله چې څلورم بیټ یی مساوي په صفر وي

شـرطیه کوډ (Condition Code) ددې په واسـطه کولای شو چې شرطیه او غیر شرطیه

لارښوونې وپیژنو.

I بیټ که چیرته $I = 0$ معنا داچې دوهم عمل وربانديې ترسره کیدونکی (Operand2) یو راجستر دی او که چیرته $I = 1$ شی معنا داچې دوهم عمل ورباندی ترسره کیدونکی (Operand2) یو سمدستي قیمت (Immediate Value) دی.

OP کوډ: ددی په واسطه د لارښوونو ډول (Types) مشخص کیږي او لاندې د معلوماتو د پروسیس کوونکو لارښوونو OP کوډونه دي

Instruction	Op Code	
AND	0000	
EOR	0001	
SUB	0010	
RSB	0011	
ADD	0100	
ADC	0101	
SBC	0110	
RSC	0111	
TST	1000	
TEQ	1001	
CMP	1010	set condition by Op1-Op2
CMN	1011	set condition for Op1+ Op2
ORR	1100	
MOV	1101	Rd=operand2
BIC	1110	
MVN	1111	Rd= NOT operand2

S بیټ که $S = 0$ نو د PSR راجستر فلګ بیټ نه بدلیږي که چیرته $S = 1$ شي نو د PSR شـرطیه فلګ بیټ تعینیږي Rn لمړی عمل ورباندی ترسره کیدونکی(Operand) دی چې د R0 څخه تر R15 پورې هر راجستر پکی راتللی شي.

Rd دا منزل مقصـود یـا هدف (Destination) راجـسـتر دی چې د R0 څخه تر R15 پورې هر راجستر پکې راتللی شي.

دوهم عمـل وربـاندی ترسـره کیدونکی (Operand2) کله چې $I = 0$ نو دوهم عمل کوونکی (Operand2) یو راجستر دی چې ۱۲،۸ شکل ددی جوړښت څرګندوي.

#Shift ددی په واسطه دا مشخص کيږي چې د Rm څو ځله سمدستي قيمت (Immediate Value) لیږلی (Shift) دې.

SH ددی په واسطه د عمليو ډول مشخص کيږي.

Rm دوهم عمل وربانددی ترسره کیدونکی (Operand) دی.

<u>Operation SH Value</u>

LSL 00 Logical Shift Left

LSR 01 Logical Shift Right

ASR 10 Arithmetic Shift Right

ROR 11 Rotate Right

مثال: لاندی لارښوونی ماشين کوډ ته واړوئ.

ADD R1, R2, R3, LSL #3

31 28	27 26	25	24 21	20	19 16	15 12	11 7	6 5	4	3 0
Cond 1110	0 0	I 0	Op code 0100	S 0	Rn 0010	RD 0001	#Shift 0011	SH 00	0	RM 0011

کله چې يو څلور بيټ Operand2 قيمت يو شي نو Rm بايد يو راجستر ته شيفټ شي.

۱۳،۸ شکل د ۱۱،۸ شکل د دوهم عمل وربانددی ترسره کیدونکي جوړښت ښايي

I = 0 ، نو دوهم عمل وربانددی ترسره کیدونکی به لاندې جوړښت ولري:

11 0
Immediate Value

11 8	7	6 5	4	3 0
RS	0	SH	1	Rm

۱۳،۸ شکل د دوهم عمل وربانددی ترسره کیدونکي جوړښت ښکاره کوي په هغه صورت کې چې څلورم بيټ يې صفر وي.

۷,۸ د ستیک عملیې او لارښوونې (Stack Operation And Instructions)

د حافظې (Memory) هغه برخه چې د معلوماتو د موقت وخت د زیرمه کولو لپاره استعمالیږي له ستیک څخه عبارت ده.

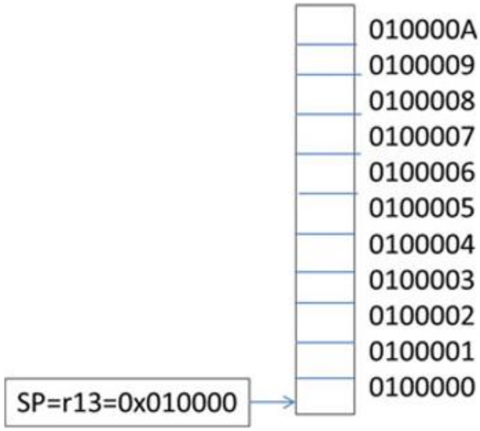
د ستیک اشاره کوونکی (Pointer) د ستیک د لمړنې برخې ادرس له ځان سره ساتي لکه په ۱۴,۸ شکل کی څرگندیږي.

R13 راجستر ته د ستیک اشاره کوونکی (Sp) ورسپارل (Assigned) کیږی او ستیک لاندی لارښوونې استعمالوي.

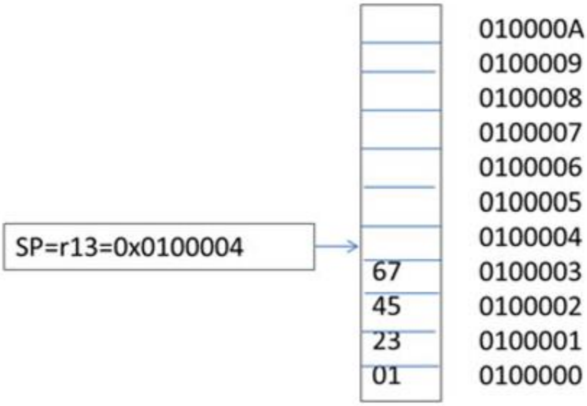
الف: Push {Condition} Rn د Rn ټوله محتوا ستیک ته لیږي او 4 د ستیک اشاره کوونکي (Sp) ته اضافه کوي.

۱۲,۸ مثال: فرض کړئ چې د R3 محتوا ۰x01234567 دی ۸,۱۵ شکل د R3 له Push کولو څخه وروسته د ستیک محتوا ښکاره کوي.

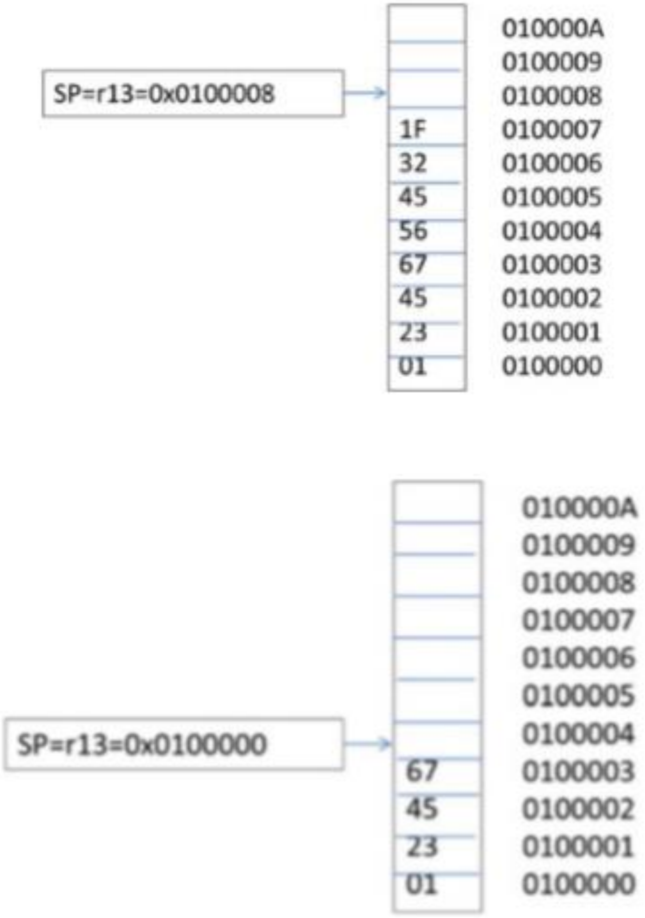
۱۴,۸ مثال: فرض کړئ چې د R4محتوا ۰X5645321F دی ۱٦,۸ شکل د R4 له Push کولو څخه وروسته د ستیک او د ستیک اشاره کوونکي (SP) محتوا ښایي.



۱۳,۸ انځور د ستيک جوړښت



۱۴,۸ انځور ستيک وروسته له push عمليی څخه ښايي



۱۵،۸ انځور د POP عملیې څخه وروسته د ستیک محتویات

POP لارښوونه: دغه لارښوونه لاندی جوړښت لري:

POP{condition} Rn

POP Rn: د Pop لارښـوونه د ستیک له لمړني برخې څخه لغات (Word) لیرې کوي او په Rn راجستر کی یی ذخیره کوي همدارنگه په اتومات ډول د ستیک اشاره کوونکې

(SP) د 4 په اندازه کمیږي.

۱۵,۸ مثال: د POP R0 لارښوونو له ترسره کیدو څخه وروسته ۱٦,۸ شکل د ستیک او د ستیک اشاره کوونکي (SP) محتویات ښايي چې د R0 متحوا به ۰x1FAD7856 وي او ستیک به د ۱۷,۸ شکل په څیر وي.

۸,۸ برینچ او برینچ له لینک لارښوونوسره Branch (B) and (Branch With Link) Instruction

د برینچ (Branch) لارښوونی لاندی عمومي جوړښت لري:

B{condition} Label
B Label ; Branch To Location Label.

B label: که چیرته فلګ بیټ د Z=1 شي نو د BEQ لارښوونې به ترسره کیږي.

BL Subroutine:
BL Subroutine; it will branch to subroutine and save contents of PC (R15) to R14 (link register) for return from subroutine

۱٦,۸ مثال: فرض کړئ چې R1 holds Y او R2 holds X داسي (Subroutine) ولیکي چې د $Y = 16X + 4$ قیمت شي؟

```
BL Funct
Funct SUB R1, R1, R1
ADD R1, R1, R2, LSL4
ADD R1, R!, #04

MOV R15, R14 ;  Move return address to PC
```

31 — 28	27 — 25	24	23 — 0
Cond	101	L	offset

۱،۸،۸ **د B او BL لارښـوونی جوړښـت:** $L = 0$ معنا د برینچ او شرط د برینچ لپاره چې د Cond field په واسطه تعین کیدای شي. $L = 1$ معنا د برینچ او لینک دی.

لارښوونې *(Instruction):*

BAL Branch Always
BEQ Branch if Equal
BNE Branch if Not equal
BPL Branch on positive
BMI Branch on negative
BCC Branch if carry flag is clear
BLO Branch below for unsigned number
BCS Branch carry flag is set
BHS Branch if higher for unsigned number
BVC Branch if Over flow flag is clear
BVS Branch if Over flow flag is clear
BGT Branch greater for signed number
BGE Branch greater or equal for signed number
BLT Branch Less than for signed number
BLE Branch Less than for signed number
BLS Branch less than or equal for unsigned number

۱۷،۸ مثال: لاندی د اسمبلي ژبې کوډ د شرطیه لارښوونو پر بنسټ ولیکئ؟

$CMP\ R1, R2$
$BEQ\ Exit$
$ADD\ R1, R2, R3$
$Exit:$
$SUB\ R1, R5, R6$

د پورتنئ اسمبلي ژبي کوډ د شرطیه لارښوونو په شکل په لاندې ډول دی:

$CMP\ R1, R2$
$SUBEQ\ R1, R5, R6$

$ADDNE\ R1, R2, R3$

۹,۸ – *Multiply (MUL) and Multiply Accumulate (MLA) Instructions*:

MUL instruction
MUL Rd, Rm, Rs ; Rd = Rm ∗ Rs
MLA Multiply and Accumulate

MLA Rd, Rm, Rs, Rn ; Rd = Rm ∗ Rs + Rn

۱,۹,۸ د ضربي لاربښوونو جوړښت (Multiply Instruction Format):

31 28	27 22	21	20	19 16	15 12	11 8	7 4	3 0
Cond	00000	A	S	Rd	Rn	RS	1001	Rm

$A = 0$ *MUL instruction*
$A = 1$ *MLA instruction*
$S = 0$ *Do not change flag bit*
$S = 1$ *Set the flag bits*
Rd is destination register

Rs, Rm and Rn are the operands

۱۰,۸ خلاصه

- د کمپایلر یوه دنده دا ده چې HLL اسمبلي ژبې ته تبدیل کړي.
- د اسمبلر دنده دا ده چې اسمبلي ژبه ماشین (Binary) ژبې ته تبدیل کړي
- د کمپیوټري لارښـوونې د نیمونیک (Mnemonic) پواسـطه ښـودل کیږي لکه "ADD."
- هره لارښـونه شـاید یو یا دوه یا درې عمل وربانډی ترسـره کیدونکي ولري، ADD R1, R2او R3 چیرته چې R1, R2, او R3ته عمل وربانډی ترسره کیدونکی وایي.
- ARM پرمختللی RISC پروسیسر ماشین ته وایي او ARMv7 بیا 32 بیټه او 16 بیټه لارښوونې استعمالوي.
- ARMv7 ټول 31 راجسترونه لري چې یوازې 16 راجستره د R0 تر R15 پورې د پروګرامر په واسطه استعمالیږي.
- د R15 راجسـتر د پروګرام د شـمیرلو لپاره اسـتعمالیږي (PC)، R14 بیا د لینک راجستر لپاره استعمالیږي (LR)، او R13 د سټک فوانټر لپاره استعمالیږي(SP).
- د PSR راجسـتر د کنټرول بیټونو لپاره اسـتعمالیږي (I, F, M, and T) او د پلګ بیټونو لپاره.,N, Z , C and V)
- د ARM پروسیسر دوه قسمه لارښوونې وړاندې کوي، او دا دواړه بیا مشروط او غیر مشروط لارښوونې دي.
- د CMP, CMN, TST, and TEQ لارښـوونې به د موقیعت راجسـتر لپاره ځای په ځای شي.
- د حافظې یوه حصـه د سـټک لپاره استعمالیږي، او سټک فوانټر د سټیک د پورتنی حصې ادرس ساتي.
- نهم چپټر د ARM زیـاتې لارښـوونې پـه بر کې نیسـي لکـه لوډ، سـاتل، سـوډو لارښـوونې، د بیټ د سـاحې لارښوونې، ARM ادرس طریقې، او د معلوماتو تصویر په حافظه کې.

مشکلات او پوښتنې Problems and Questions

۱. تشريح کړئ چې څرنګه لوړ کيفيت ژبې کوډ HLL ماشين کوډ ته تبديليږي

۲. د عمل وربانديی ترسره کيدونکي د تعداد په اساس د لارښوونو اقسام تشريح کړئ.

۳. د ARM پروسيسر کوم راجستر د پروګرام شميرونکی PC لپاره استعماليږي؟

۴. د ARM پروسيسر کوم راجستر د ستيک نښه کوونکی SP لپاره استعماليږي؟

۵. د ARM پروسيسر کوم راجستر دلينک راجستر لپاره استعماليږي؟

۶. د لاندی لارښـوونو د اجرا نه وروسته د R5 محتويات به څه وي، فرض کړئ R2 کې 0X34560701 دي او R3 کې 0X56745670 دي.

- ADD R5, R2, R3
- AND R5, R3, R2
- XOR R5, R2, R3
- ADD R5, R3, #0x45

۷. فرض کړئ چې R2 =0x00001234. د R1 محتويات به څه وي ؟

- MOV R1, R2, LSL #4
- MOV R1, R2, LSR #4

۸. د لاندې دواړو لارښوونو ترمنځ فرق څه شی دی؟

- SUBS R1, R2, R2
- SUB R1,R2, R2

۹. لاندې HLLژبه ARM لارښوونو ته تبديل کړئ.

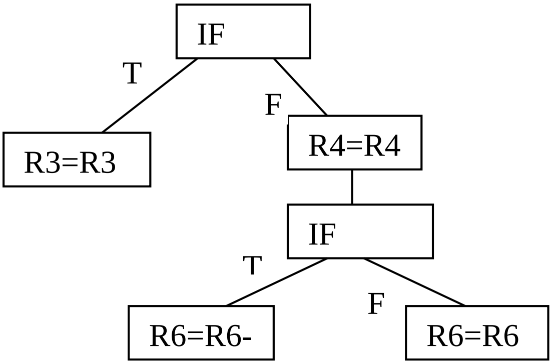
```
IF R1>R2 AND R3>R4 then
R1= R1 + 1
Else
R3=R3 + R3*8
Endif
```

۱۰. لاندې HLLژبه ARM لارښوونو ته تبديل کړئ.

```
IF R1>R2 OR R3>R4 then
R1= R1 + 1
Else
```

R3=R3 + R5*8
Endif

۱۱. لاندې فلو چارټ ARM اسمبلي ژبې ته تبدیل کړئ.



۱۲. داســـې یو پروګرام ولیکئ چې 0 تر 10 عـددونـه واخلي یـا لاندې د C ژبې پروګرام ARM اسمبلي ژبې ته تبدیل کړئ.

```
int sum;
int i;
sum = 0;
for (i = 10 ; i > 0 ; i - - ){
sum = sum + 1
}
```

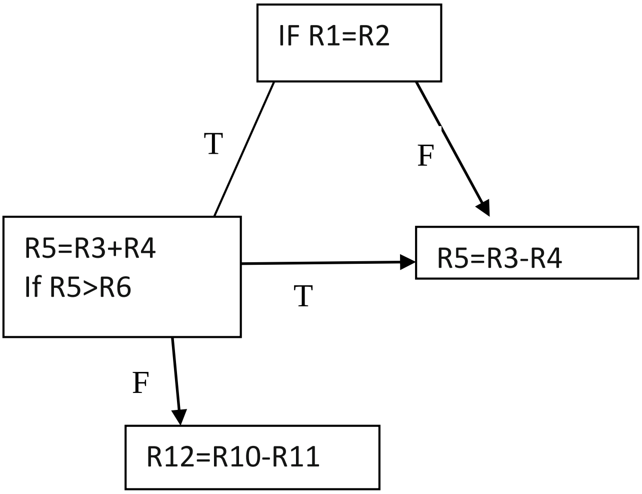
۱۳. داسې پروګرام ولیکئ چې لاندې HLL ژبه ARM لارښوونو ته تبدیل کړي.

```
a= 10;
b=45;
while ( a! =b ) {
if (a < b)
a = a + 5;
else
b= b + 5;
}
```

۱۴. لاندي HLL ژبه ARM لارښوونو ته تبدیل کړئ.

IF R1>R2 AND R3>R4 then
R1= R1 + 1
Else
R3=R3 + R5*8
Endif

۱۵. لاندي فلو چارټ ARM اسمبلي ژبې ته تبدیل کړئ؟



نهم څپرکی

د ارم (ARM) لارښوونې دوهمه برخه
(ARM Instructions Part II)

موخې: د دې څپرکې په پای ته رسیدلو سره به تاسی وکولای شئ چې:

- د لارښوونو د لوډ (Load Instruction) بیلابیل ډول لارښوونې تشریح کړای شي.
- د زیرمه کولو لارښوونې (Store Instruction) بیلابیل ډولونه لیست کړای شئ.
- د ای آر ام (ARM) ادرس ورکولو مختلفو ډولونو کې توپیر وکړائ شئ.
- د ای آر ام (ARM v7) ورژن 7 مشابه (Pseudo) لارښوونې لیست کړائ شئ.
- د ADR او LDR لارښوونو پلي کول زده کړئ.
- د ساحې لارښوونې لارښود عملیات توضیح کړائ شئ.
- زده کړئ چې معلومات په حافظه کې څنګه ښکاره کیږي.

۱،۹ پیژندنه

د معلوماتو لیږدولو لارښـــوونې د حافظې څخه راجســـترونو او راجســـترونو څخه حـافظې ته د معلوماتو د لیږدولو له پاره کارول کیږي. د ای آر ام (ARM) پروســـیســـر حافظې ته د لاسـرسـي له پاره LDR او STR لارښـوونې کارويLDR او STR ددی وړتیا لري چې په غیرمسـتقیمه توګه ، حافظې ته د لاسـرسـي له پاره دمخه شـاخص ادرس (Pre-Index-Addressing)، او وروســـتــه شـــاخــص ادرس (-Post-Index Addressing) وکاروي . ای آر ام (ARM) یو شـمیر سـوډو لارښـوونې وړاندې کوي چې د پروګرام لیکونکې (Programmer) او اســـمبلر لخوا کارول کیږي ترڅو دا د ای آر ام (ARM) لارښوونو ته تبدیل کړي.

۲،۹ د ای ار ایـم د مـعـلومـاتو د لیږد لارښـــوونې (ARM Data Transfer Instructions)

د لوډ لارښوونې LDR: LDR د لارښوونې د حافظې څخه د معلوماتو لوستلو له پاره کارول کیږي او په راجستر کې یې ذخیره کوي ، او دا لاندې عمومي جوړښت لري.

$$LDR[type]\{condition\}\ Rd, Address$$

چیرې چې (Type) لاندې لوډ لارښوونې (Load Instruction) تعریف کوي.

Ldr	Load 32 Bits (Word)
Ldrb	Load 1 Byte
Ldrh	Load 16 Bits (Half Word)
Ldrs	Load Signed Byte
Ldrsb	Load Sign Extension
Ldrsh	Load Signed Half Word
Ldm	Load Multiple Words

دلته یو اختیاري حالت دی داسې چې LDREQ د معلوماتو لوډ (Load Data) که چیرې د Z فلګ یو وي او Rd د منزل مقصود راجسټر (Destination) وي

۱،۹ مثال: فرض کړئ $R0$ د 0000 ادرس لري او لاندې حافظه ورکړل شوې ، د لارښوونو اجرا کولو وروسته د R1 او R3 محتوا ښایې.

Address	Contents
0	0X85
1	0XF2
2	0X86
3	0XB6

```
LDRH R1, [R0]      R1 = 0x0000F285
LDRSH R3, [R0]     R3 = 0xFFFFF285
```

۱،۲،۹ د آی ارام مشابه لارښوونې (ARM Pseudo Instructions): ای آر ام (ARM) د ډیری مشابه (Pseudo) لارښوونو د حمایت قابلیت لري؛ د جعلي لارښوونې د برنامه لیکوونکي لخوا کارول کېږي ، او اسمبلر مشابه (Pseudo) لارښوونې د ای ار ام لارښوونو ته اړوي.

د *ADR مشابه لارښوونې:* ADR د حافظې د ځایونو ادرس په راجسټرونو کې اچوی او لاندې جوړښت و لري.

$$ADR\ Rd, Address$$

۲،۹ مثال: لاندی لارښوونې د ډیټا ادرس لولي اوبیا یی په R3 راجستر کی اچوي.

ADR R0, Table	Move Address Represented By Table
LDR R3, [R0]	R3 ¼ 0x23456780

Address	Data
Table	0x23456780

د LDR *مشابه لارښوونې:* LDR مشابه لارښوونې په راجسټر کې د ثابت اچولو له پاره کارول کېږي د دی له پاره چې د ۳۲ بیټونو ثابت انتقالول راجسټر ته دغه لارښوونه $MOV\ Rd, \#value$ کولای شي یواځې ۱۲ بیټونه Rd راجسټر ته انتقال کړي ځکه چې دوهم عمل کوونکی $MOV\ 12\ bits$ ارشوونو بڼه ده. د LDR مشابه لارښوونې لاندې بڼه لري.

۹,۳ مثال: لاندې لارښوونه به د $R1$ په راجسټر کې 0x23456789 واچوئ

LDR R1, =0x23456789

۲,۲,۹ د ذیرمه کولو لارښوونې (Store Instruction): د STR لارښوونې ددې له پاره استعمالیږي تر څو د راجسټرونو محتویات حافظې ته ولیږدوي او لاندې عمومی بڼه لري

$$\text{STR[Type]}\{condition\}\ \text{Rd}, [address]$$

چېرې چې "Type" لاندې لارښوونو ډولونه ټاکي:

Store 32 Bits (Word)	STR
Store 1 Byte	STRB
Store 16 Bits (Half Word)	STRH
Store Multiple Words	STM

۴,۹ مثال: STR R5, [R3] چې د حافظې د کوم موقعیت ادرس له ځان سره لري په هغې کې د R5 محتویات ذیرمه کړئ: R3 اوس بنسټی راجستر دی.

۳,۹ د ای ار ام ادرس ورکولو طریقې (ARM Addressing Mode)

د ای آر ام (Arm) پروسیسر ډیری د ای آر ام (Arm) وړاندیزونو ادرس

ورکولو طریقې حمایه (Support) کوي او دوی دمخه ترتیب شــوي چې پدې ترتیب دي: مخکېني شاخص له سمدستي افسیټ سره ، د مخکېني شاخص د راجستر افسیټ سره، د مخکیني شاخص د اندازه شوي راجستر سره ، مخکیني شاخص د راجستر له افسیت او بیا لیکلو (Write Back) سره
وروستی شاخص له سمدستی افسیټ سره وروستی شاخص د راجستر له افسیټ سره وروستی شاخص له اندازه شوي راجستر سره.

لاندې جدول د ای آر ام (ARM) د ادرس حالتونه ښکاره کوي:

Addressing Mode	Assembler Syntax	Effective Address (EA)
Immediate	MOV R1, #0X25	Data Is Part Of Instruction
Pre-Indexed	[Rn]	EA ¼ Rn
Pre-Indexed With Immediate Offset	[Rn, # Offset]	EA ¼ Rn + Offset
Pre-Indexed With Register Offset	[Rn, T Rm]	EA ¼ Rn T Rm
Per-Index With Scaled Register	[Rn, Rm, Shifted]	EA ¼ RN + Rm Shifted
Pre-Indexed With An Immediate Offset And Write Back	[Rn, Offset]!	EA ¼ Rn + Offset Rn ¼ Rn + Offset
Addressing Mode	Assembler Syntax	Effective Address (EA)
Pre-Index With Register Offset And Write Back	[Rn, T Rm,]!	EA ¼ Rn T Rm Rn ¼ Rn T Rm
Pre-Index With Scaled	[Rn, Rm, Shifted]!	EA ¼ Rn T

Register Offset And Write Back		Rm Shifted Rn ¼ Rn T Rm Shifted
Post-Index With Immediate Offset	[Rn], Offset	EA ¼ Rn Rn ¼ Rn + Offset
Post-Index With Register Offset	[Rn], T Rm	EA ¼ Rn Rn ¼ Rn T Rm
Post-Index With Scaled Register Offset	[Rn], T Rm. SHL #N	EA ¼ Rn Rn ¼ Rn T Rm Shifted

۱,۳,۹ سـمدسـتي ادرس ورکول (Immediate Addressing): په سـمدسـتي ادرس ورکولو کې عمل ورباندې ترسره کيدنکي د لاربښونو يوه برخه ده.

MOV R0, # 0x34

يا:

ADD R1,R2, 0x12

۲,۳,۹ مخکې ترتيب شـوی (Pre Indexed): په مخکښي شـاخص کې د ادرس ورکولو حالت د[Rn] په واسطه ښودل کيږي د [Rn] محتوا (EA) يا موثره ادرس دی لکه:

LDR R2, [R3]

د سـمدسـتی آفسـيټ سره دمخه ترتيب شـوی (Pre-Indexed With Immediate Offset)

د سـمدسـتي آفسـيټ سره دمخه شاخص شوي [Rn, #Offset] په واسطه ښکاره کيږي لکه:

LDR R0, [Rn, #Offset]

آفسيټ يو سمدستي قيمت دی لکه:

$$LDR\ R1, [Rn, \#025]$$
$$EA = R2 + 0x25$$

د راجستر آفسيټ سره دمخه ترتيب شوي (Pre-Indexed With Register Offset)

افسيټ کېدلی شي راجستر يا هغه راجستر وي چې د لغت چپ او ښي طرف ته تلونکې عمليه ولري

$$LDR\ R0, [R1, R2]$$
$$EA = R1 + R2$$

۵,۹ مثال: د لاندې ادرسونو نه مؤثره ادرس کوم يو ده؟ فرض کړئ R5 د 0X00002345 محتوا لري:

$$[R5, \#0x25]$$
$$EA = 0x000002345 + 0x25 = 0x0000236A$$

٦,۹ مثال: د لاندې شاخص ادرسونو موثره ادرس کوم يو ده، فرض کړئ د R5 0x00001542 محتوا او 0X00001000 د R2 محتوا ده

$$[R5, R2]$$
$$EA = R5 + R2 = 0X00001542 + 0X00001000$$
$$= 0X00002542$$

د اندازه شوي راجستر سره مخکښې ترتيب شوی:

آفسيټ د لغت چپ او ښي طرف ته تلونکي عمليو سره راجستر لري.

LDR R0, [Rn, R2, LSL#2]

۷,۹ مثال: د لاندې لارښوونو موثره ادرس څه شی دی؟

LDR RO, [Rn, R2, LSL#2]

$EA = RN + R2 * 4$

پورته R2 دوه ځله چپ طرف ته اړول شوی بیا (د 4 سره ضرب شوی) او Rn ته اضافه شوی.

۳,۳,۹ د بیرته لیکلو سره مخکې نه ترتیب شوی Pre-Indexed With Write (Back): عمومي جوړښت د مخکې نه ترتیب شوی ادرس ورکول د بیرته لیکلو سره.

[Rn, Offset]!

د حیرانتیا (!) کرکټر د بیرته لیکلو له پاره کارول کیږي؛ آفسیټ کیدای شي سمدستي قیمت یا راجستر وي یا لیږدول شوی راجستر وي:

$EA = Rn + offset\ and\ Rn = Rn + offset$

د سمدستی آفسیټ سره مخکې نه ترتیب شوی او بیرته لیکلو.

$LDR\ R0, [R1, \#\ 4]!$

$EA = R1 + 4$ او $R1$ بیا ځلې نوی شوی $\mathrm{R1} = \mathrm{R1} + 4$

۸,۹ مثال: د لاندې لارښوونو له پاره د R5 مؤثره ادرس او نهایي قیمت څه دی؟ د $R5 = 0x\ 00002456$ محتوا فرض کړئ:

$LDR\ R0, [\ R5; \#0X4\ ]!$

$EA = R5 + 0x4 == 0x000245A$

$R5 = R5 + 0x4 == 0x000245A$

د راجستر آفسیټ او بیرته لیکلو سره دمخکې نه ترتیب شوی.

۹,۹ مثال: د لاندې لارښوونو R5 مؤثره ادرس او نهایي قیمت څه شی دی؟ د R5 0x

00002456او R2 0X00002222 محتوا فرض کړئ:

LDR R0, [R5, R2]!

$EA = R5 + R2 = 0x00004678$

$R5 = R5 + R2 = 0x00004678$

د اندازه‌شوي راجستر آفسيټ او بيرته ليکلو سره دمخکې نه ترتيب شوی.

LDR R1 ، [Rn ، R2 ، LSL # 2]!

$EA = Rn + R2 * 4$

$Rn = Rn + + R2 * 4$

۴,۳,۹ د وروسته ترتيب شوي ادرس ورکول (Post-index Addressing): د

وروسته ترتيب شوي ادرس ورکول عمومي جوړښت په لاندې ډول دی:

DR R0, [Rn], Offset

آفسيټ(Offset) کيدای شي سمدستي قيمت يا راجستر يا شفټ شوی راجستر (Shift Register) شي.

د سمدستی قيمت سره وروسته ترتيب شوی (Post-Index With An Immediate Value

LDR R0, [Rn], #4

$Effective\ Address = Rn\ and\ Rn + Rn + 4$

د راجستر آفيسټ سره وروسته ترتيب شوی (Post-Index With Register Offset)

LDR R0, [Rn], Rm

$, Effective\ Address = Rn\ and\ Rn = Rn + Rm$

د اندازه شوي راجستر آفسيټ سره پوسټ شاخص (Post-Index With Scaled Register Offset)

LDR R0, ½Rn_, Rm, SHL#4

$Effective\ Address = Rn\ and\ Rn = Rn + Rm * 16$

۴,۹ د راجسـټر او حافظې تر منځ ادلون بدلون (Swap Memory (And Register

د مبـادلې (Swap) لارښـوونې د لوډ (Load) او ذیرمـه کولو (Store) لارښـوونې په یوه لارښـوونه کې سـره یوځای کوي او عمومي جوړښت یې په لاندې ډول دی .

SWP Rd, Rm, [Rn]

په پورتنی جوړښـت کې Rd د منزل مقصـود (Destination) راجسـټر Rm د مبادلې (Swap) حافظه او Rn اساسي راجسټر (Base Register) دی.

راجسـټ آر ډي د منزل راجسـټریشـن ، د Rm سـویپ حافظه او راجسـتر (SWAP) د سرچېنې راجستر دی، او Rn اساس راجستر دی.

د مبادلې (Swap) لارښوونې لاندې عملیې تر سره کوي

د تبادلې لارښوونې لاندې دندې ترسره کوي.

Rd← Memory [Rn] Load Rd From Memory Location [Rn]
[Rn]← Rm Store The Contents Of Rm In Memory Location [Rd]
SWPB Rd, Rm, [Rn] Swap One Byte

۵,۹ د بیټونو ساحوي لارښوونې (Bits Field Instructions)

ای ار ام (ARM) دوه ډوله د سـاحې لارښـوونې (Field Instruction) وړاندې کوي Bit Field Clear (BFC) د Bit Field Insertion(BFI)

۱,۵,۹ ساحوی بیټونو روښانه لارښوونې (Bit Field Instruction(BFC:

BFC عمومي جوړښت په لاندې ډول دی:

$$BFC\ \{cond\}\ Rd, \#\ lsb, \#width$$

Rd: منزل مقصود راجستر (Destination Register) دی.

Lsb: په منزل مقصود راجستر کې (Rd) د بیټ د موقعیت شروع روښانه کوي.

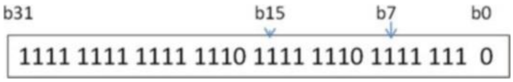
Width: د Lsb نه Msb ته د Rd راجستر د پاکولو له پاره د بیټونو تعداد روښانه کوي.

۹. ۱۰ مثال: داسې لارښووني ولیکئ چې 7 تر 15 د R4 راجستر بیټونه پاک کړي:

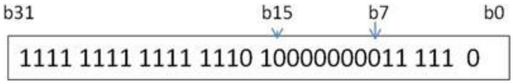
فرض کړئ R4 کې 0xFFFEFEFE قیمت موجود دی.

BFC R4, #7, #8 clear bit 7 through bit 15 (8 bits) of register R4.

ابتدايي قیمت په R4 کې:



په R4 کې ورسته له پاکولو د 7 بیټ نه تر 15 بیټه پوری نتیجه به داسې وي:



۲,۵,۹ د بیټ ځایولو لارښووني (Bit Filed Instruction): د بیټونو ځایولو لارښووني (BFI) د دی له پاره استعمالیږي چې د بیټونو یو سیټ له یوه راجستر RN څخه یې بل راجستر RD ته کاپی کړي کوم چې شروع له Lsb د Rd ته کوي: BFI لاندې بڼه اختیاروي.

BFI{cond} Rd, Rn, #lsb, #width

RD: منزل مقصود راجستر دی.

Rn: د منبع راجستر دی.

#lsb: شروع د بیټونو ده RN ته.

#width: د بیټونو تعداد دی کوم چې له Lsb نه شروع کیږي RN ته

۱۱,۹مثال: له R3 څخه 8 بیټونه چې د 4 څخه شــروع R4 ته کاپی کړئ؛ فرض کړئ R3د $0x\ FFFFEBCD$ محتوا لري او R4 پکې $0xEE035007$ قیمت لري.

Bfi R4, R3, #4, #8, او دا لارښوونې به د B4 څخه له 8 بیټ نه د R3 تر B11 په B0 کې له B7 تر R4 کاپی کړي،

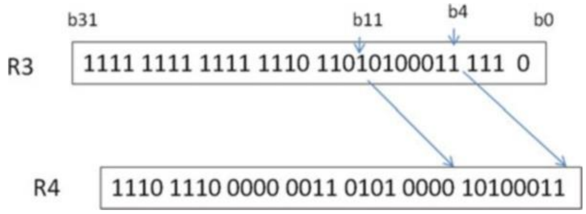
ابتدایي قیمت د R3 په دوه ییز کې:

11101110000000 11010100000000111

په دوه ییز کې د R4 لومړنی قیمت دی:

11101110000000 11010100000000111

دا لارښـوونې به 8 بیټه د 4 بیټ نه د R3 په R4 کې چې شروع یي له 0 نه تر R4 پورې وي کاپی کړي



٦,٩ د معلوماتو ښکاره کول او حافظه

(Data Representation And Memory)

د ای ار ام (ARM) پروسیسر یوه کلمه (Word) د څلور بایټونو په ډول تعریفوي او د کلمې نیمایي یي ۲ بایټونه دي معلومات کېدای شي د ۱٦ په قاعده، د ۱۰ په قاعدې او یا د ۲ په قاعدې ښکاره کړئ

a. د ۱۰ په قاعدې شميرې لکه ۳۴۵

b. د ۱٦ په قاعدې شميرې لکه $0x2345$ چې x د ۱٦ قاعده ښکاره کوي

c. د ۲ په قاعده لکه ۱۰۱۱۱۱۰۰

حافظه معلومات او کوډ له ځان سره ساتي ۱۱،۹ شکل د حافظې يو بلاک ډياګرام ښايي. د حافظې ادرس د معلوماتو موقعيت ټاکي ، چې د ای ار ام (ARM) پروسسر حافظه هرموقعيت يو بايټ لري، په اسمبلي ژبه کې ، يو ليبل (Label)، لکه څنګه چې په ۲،۹ شکل کې ښودل شوی د حافظې ادرس ښکاره کوي.

۲،۹ شکل ښايي چې څنګه د هرې حافظې موقعيت يو بايټ ساتي. د دوه بايټونو (نيم لغت) معلوماتو ذخيره کوي ، لکه $0x4563$، په دوه مختلفو لارو لوی انډيان (Big Endian) او کوچني انډيان (Little Endian) ساتلی شو

لوی انډيان (Big Endian). په لوی انډيان کې د معلوماتو خورا مهم بايټ (MSB) لومړی په حافظه کې ذيرمه شوی وي. په ۳،۹ شکل کې، د $0x4563$ له پاره 45 ترټولو د پام وړ بايټ دی ، او 63 لږ تر لږه د پام وړ بايټ دی.

د ای ار ام (ARM) ورژن 7 حافظې هر موقعيت يو بايټ لري او يوه کلمه (4 بايټونه) په دوه بيلابيلو (لوی انډيان او کوچنی انډيان) په وسيله په حافظه کې ساتل کيدای شي:

لوی انډيان: په لوی انډيان کې د يوې کلمې خورا مهم بايټ په ټيټ ادرس کې ذيرمه شوي وي.

۱۲،۹ مثال: د $0x34569312$ ممکن د لوی انډيان په شکل کې وساتل شي لکه څنګه چې په ۳،۹ شکل کې ښودل شوي.

00	23
01	4A
10	56
11	F5

۱،۹ انځور حافظه له دوه ييز (Binary) ادرس سره

List	23
List+1	4A
List+2	56
List+3	F5

۲,۹ انځور حافظه د پیژندنی (Labeling) له ادرس سره

Memory

000	34
001	56
010	93
011	12

۳,۹ انځور ۳۴۵۶۹۳۱۲ نمبر د شپاړلسو قاعدې لوی انډیان ښکاره کول

Memory

000	12
001	93
010	56
011	34

۴,۹ انځور: ۳۴۵۶۹۳۱۲ نمبر د شپاړلسو قاعدې کوچنی انډیان ښکاره کوي

کوچنی انډیان: په کوچني انډیان کې دیوی کلمې لږترلږه د مهم بیټ په ټیټ ادرس کې زیرمه شوی وي.

۱۳,۹ مثال: د 34569312 د ۱٦ قادی شمیره ممکن په کوچني اندیان شکل کې وساتل شي لکه څنګه چې په شکل ۴,۹ کې ښودل شوي.

۷,۹ خلاصه

د ای ار ام (ARM) لارښوونې حافظې ته لوستلو او لیکلو له پاره LDR او ST کاروي.
د لوډ (Load) لارښـوونو لارښود د یو بایټ لوډ (LDRB) ، 2 بایټ لوډ (LDHB) او د 4 بایټونولوډ (LDR) پورته (Load) کولو له پاره کارول کیدای شي.
LDRSB (Load Signed Extension) د یوې بـایـټ لوډ (Load) کولو لـه پـاره استعمالیږي او د معلوماتو د نښه کولو له پاره دوام پیدا کوي.
LDRSH (Load Signed Extension) د دوه بـایټونو لوډ (Load) کولو له پاره استعمالیږي او د معلوماتو د نښه کولو له پاره دوام پیدا کوي.
د ARM مشابه (Pseudo) لارښوونې ADR (د حافظې د موقعیت د ادرس را اوچتول) او LDR (راجستر ته د 32 بیټه قیمت انتقالون) ده.
د ای ار ام (ARM) پروسـسـر ذخیره کولو (Store) لارښـوونې عبارت دي له STR د (یوی کلمی ذخیره کول) STRB (یو بایټ ذخیره کول) STRH (د کلمی نیمایي ذخیره کول) دي.
ای ار ام (ARM) مختلف ادرس ورکولو طریق وړانـدې کوي لکـه: مخکې نـه ترتیـب شـوی، مخکې نه ترتیب شـوی د سـمدستی افسټ سره، مخکې نه ترتیب شوي د راجستر افسټ سره، مخکې نه ترتیب شوی د متوازن راجستر سره، مخکې نه ترتیب شوی د افسټ راجسـتر او بیرته لیکل شـوي افسـټ سـره، وروسـته ترتیب شـوی د سـمدستی افسټ سره، وروسـته ترتیب شوی راجستر افسټ سره، او وروسته ترتیب شوی د متوازن راجستر افسټ سره.
معلومات د لوی انډیان او کوچني انډیان په شکل کې په حافظه کې وړاندې کېدای شي.
په لوی انډیان کې د کلمې خورا مهم بایټ په ټیټ ادرس کې زیرمه شوي وي.

په کوچني انډیان کې د کلمې لږترلږه د پام وړ بایټ په ټیټ ادرس کې زیرمه کېږي.

مشکلات

1. د ارم لوی انډیان په کارولو ســره لاندې لارښــوونې ومومئ؛ فرض کړئ لیســت د حافظې له $0x0000018$ ځای څخه پیل کېږي

ADR R0, LIST ; Load R0 With Address Of Memory Location list

MOV R10, #0x2

(a)LDR R1, [R0]

(b) LDR R2, [R0, #4]!

(c)LDRB R3, [R0], #1

(d)LDRSB R5, [R0], #1

(e)LDRSH R6, [R0]

$LIST\ DCB 0x34, 0xF5, 0x32, 0xE5, 0x01, 0x02, 0x8, 0xFE$

2. د لمړنی مشکل په A او B برخه کوچنی انډیان په استعمالولو سره کار حل کړئ.

(a) $R1 = \ 0xE532F534$

(b)R2 $= 0xFE080201$

3. لاندې پروګرام د چلیدو څخه وروسته به د R7 راجستر محتوا څه وي.

$ADR\ R0, LIST$

$LDRSB\ R7, [R0]$

$LIST\ DC\ 0xF5$

4. د لاندې لارښوونو د لوړ له پاره به د RI راجستر محتوا څه شی وي فرض کړئ R0 د کوچنی انډیان سره د لیست ادرس لري.

(a) $LDR\ R1, [R0]$

(b) $LDRH\ R2, [R0]$

(c) $LDRB\ R3, [R0], \#1$

(d) $LDRB\ R4, [R0]$

(e) $LDRSB\ R5, [R0], \#1$

(f) $LDRSH\ R6, [R0]$

$List\ DCB 0x34, 0xF5, 0x32, 0xE5, 0x01, 0x02$

5. لاندې حافظه درکړل شــوی ده چې د هر راجســتر محتوا ښــايي، فرض کړئ چې د $R1 = 0x0001000$ او $R2 = 0x00000004$ کوچني انډيان په استعمال سره.

(a). $LDR\ R0, [\ R1]$
(b). $LDR\ R0, [R1, \#4]$
(c) $LDR\ R0, [R1, R2]$
(d) $LDR\ R0, [R1, \#4]!$

1000	23
	13
	56
	00
1004	45
	11
	21
	88
1008	03
	08
	35
	89
100C	44
	93

6. د لاندې لارښوونو د ترسره کېدو څخه وروسته به موثره ادرس او د R5 محتوا څه وی؟ فرض کړی R5 محتوا R0 18 ده او د R6 محتوا .$X00000020$۰ ده.

$(a)\quad STR\ R4, [R5]$
$(b)\quad STR\ R4, [R5, \#4]$
$(c)\quad STR\ R4, [R5, \#8]$
$(d)\quad STR\ R4, [R5, R6]$
$(e)\quad STR\ R4, [R5], \#$

لسم څپرکی

ای ار ام (ARM) اسمبلي ژبي پروګرام لیکل د کېل (Keil) کاریال په کارونې سره

ARM Assembly Language Programming Using Keil Development Tools

موخې: د دې څپرکي له بشپړیدو وروسته تاسو به د دې وړ یاست چې:

- د پروګرام جوړولو اوزارو دندی واضح کړای شئ
- د کراس اسمبلر (Cross-Assembler) دندي تشریح کړای شئ
- د اسمبلي ژبې د چلولو له پاره ډیری د پروګرام لیکلو پوستغالې لیستې کړای شئ
- د کېل (Keil) پروګرام لیکوونکی کاریال نصب کړای شئ.
- د پروګرم چلولو او کره کتنی (Debug) سره اشنا شئ
- د خپل پروګرام د لیکلو له پاره د مخکې نه جوړه شوې د پروګرام خاکه استعمال کړای شئ.
- د پروګرام لیکلو اصول زده کړئ.
- د اسمبلي کمپیوټري پروګرام لیکلو ژبې له پاره د کمپیوټر په حافظه کې معلومات و څرګند کړای شئ.

- د کاریال لارښونې زده کړئ.
- د معلوماتو د لارښوونو د بیلا بیلا ډولونو توپیر وکړای شئ.
- لین پـه لین پروګرام و چلولی شـی او د هر راجسـټر (Register) محتویـات مشاهده کړای شئ

۱,۱۰ پیژندنه

د پروسیسـر جوړونکي کمپنۍ د پروسیسر په باره کې معلومات خپروي چې په هغوي کې د راجسـټرونو لیستونه، د هر راجستر دنده، د معلوماتو وصلونکو (Data Bus) اندازه، د ادرس وصـلونکې اندازه، او د هغو لارښـوونو لیسـت چې اجرا کېدلی شـي. د پروسـیس مرکزي واحد (CPU) یو معلوم لارښود سیټ لري چې پروګرام لیکوونکې کولای شي چې د اسمبلي ژبې د پروګرام لیکلو له پاره یي استعمال کړئ. د لارښوونې سیټونه د هر پروسیسر له پاره ځانګړي شوی. د مثال په توګه ، د پینټیم پروسیسرونه د ای ار ام پروسیسرونو په پرتله مختلفو لارښوونو سیټ عملي کوي. هغه پروګرامونه چې د پروسـیسـر لارښـود سـیټ په کارولو سـره په اسـمبلي ژبې کې لیکل کیږي. د اسمبلر (Assembler) وظیفه ده چې د اسمبلي ژبه د ماشـین کوډ (Binary) ته واړوي تر څو د پروسـیس مرکزي واحد(CPU) یي اجرا (Execute) کړي.

کلـه چې یو اسـمبلر په یو پروسـیسـر چلول کیږي کولی شـي د مختلفو پروسیسـرونو له پاره بیلابیلې لارښـونې سـره یو ځای کړي چې دا د کراس اسـمبلر (Cross-Assembler) په نوم یادیږي. د پروسـیسـر مشـابه جوړ (Processor Simulators) د پروګرام لیکلو کلیـدي وسـیلـه ده ، ځکـه چې پـه تنظیماتو کې دوی د دې وړ شـوي چې کنټرول له پاره په ټسـټ شوي ماحول کې کـار وکړي لکـه د وینډوز یا لینکس عامل کار ځای. دوی ممکن د هدف شـوي پروسـیسـر تـه د پروګرام لېږد یا ډاونلوډ هم اسـانه کړي. لاندې د پروګرام لیکلو نمونې دي چې د ای ار ام(ARM) پروسیسرونو ملاتړ کوي.

1. ARM Keil Microcontroller Tools (WWW.Keil.Com)
2. IAR Embedded Workbench (WWW.Iar.Com)
3. GNU ARM Assembler (WWW.Gnu.Org)

۲,۱۰ کېل پروګرام لیکلو کاریال ای ار ام اسمبلي پروګرام ژبې له پاره
(Keil Development Tools For ARM Assembly)

د مثال په توګه په دې کتاب کې، Keil µVision® IDE (مدغم شوي پروګرام لیکونکې مخپانه) د میکرو کنټرولر لیکونکی سېټ 5 (MDK) ورژن استعمال شوی. کولای شي ددې پوستغالي وړیا ورژن د ای ار ام (**ARM**) ۷ ورژن د لاړښوونو ترسره کېدل چې اندازه یې له 32K څخه کمه وي چمتو او چلول یې اجرا کړي. کېل (Keil) له دې http://WWW.Keil.Com ویب سایټ څخه یې ښکته (ډاونلوډ) کولی شي.

کله چې لومړی ځل انستال شي ، نو د کڅوړو انستالر په نوم یو ډیالوګ د انستالیشن له ختمېدو وروسته خلاص شي. د غه سودمندی د کارکوونکو سره په ښکته کولو او نصبولو د µVision مختلفو تختونو او وسایلو د سمولو او اجرا کولو کې مرسته کوي. په عادي ډول ډیرې خاکې مخکې نه د µVision سره انستالیږي (۱,۱۰ شکل).

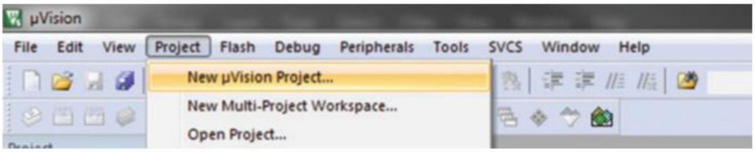
- د پیل له پاره، µVision خلاص کړئ او نوی Project انتخاب کړئ بیا وروسته ..New µVision Project انتخاب کړئ.
- خپلی پروژې ته نوم ورکړئ او د ثبت کېدو له پاره ورته په کمپیوټر کې ځای انتخاب کړئ.

د ثبت کېدو نه وروسته به یو ډیالوګ بکس خلاص شي او پیغام به درکړي چې د هدف یوه اله انتخاب کړي په دې پورې اړه لري چې ایا تاسو کوم اضافي پاکټونه د (Pack Installer) نه نصبوي یا نه، دا سکرین به ځانګړی بڼه لري. خو ای ار ام (ARM) پروسیسرونه په عام ډول نصبولوکې شامل وي. د مثال په توګه په دې کتاب کې،

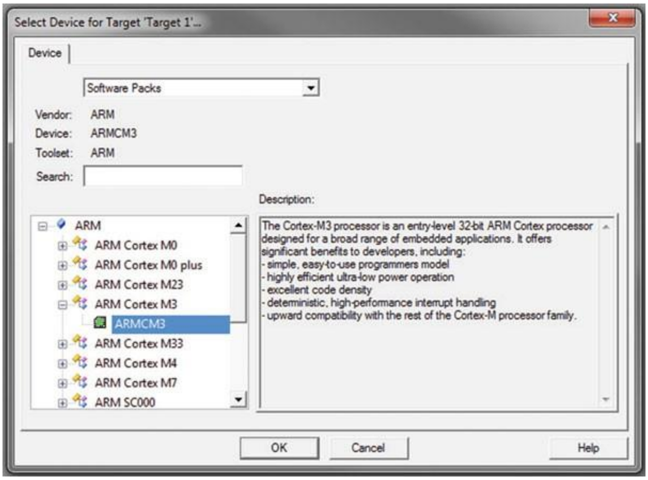
د ای ار ام کورټکس ARM Cortex M3 (ARMCM3) مخکې نه انتخاب شوی.

ARMCM3 → ARM Cortex M3 انتخاب کړئ او OK کیکاږئ (۲,۱۰شکل).

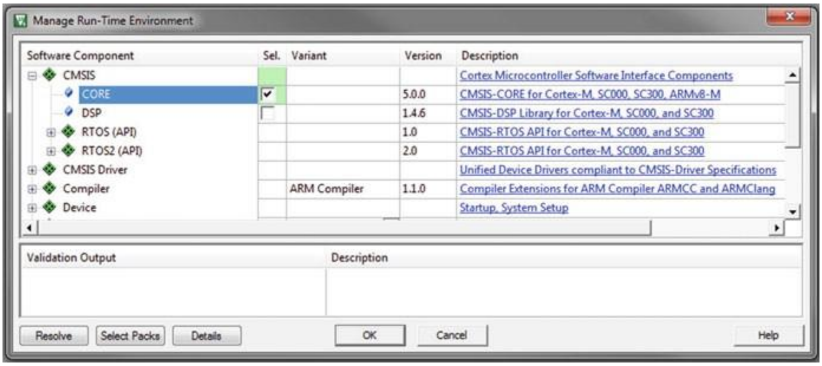
د هر پروسیسر له پاره، په µVision کې خو کومکي کتابخاني موجودي وي. بعضې يي ضروري وي، لکه د پیل د ترتیب له پاره، په داسې حال کې چې نور يي بیا د فعالولو له پاره انتخابي وي.



۱,۱۰ انځور: په کیل Keil V5.22 کی د نوي پروژی پرانستل



۲,۱۰ انځور: د ای ار ام (ARM)کورټکس M3 پروسیسر انتخابول

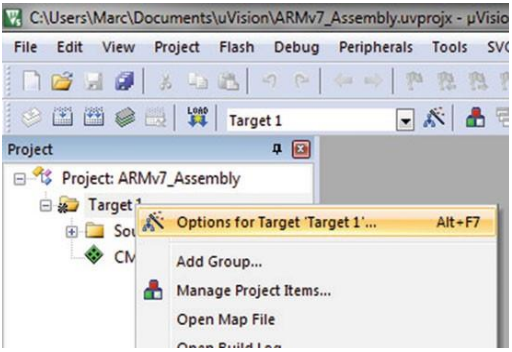


۳,۱۰ انځور: د ARMCM3 پروسيسر چلولو له پاره د CORE کومکي کتابتون

پراخه اسانتياوی، لکه انټرنيټ ډريواري او ګرافيکس مخينه، د ARMv7 اسمبلي پروګرام عادي چلولو له پاره، يوازې د شروع برخي اجزاوي ضروري وي.

- CMSISCORE انتخاب کړئ او بيا OK کېکاږئ (شکل. (۳,۱۰)

اوس، بايد تاسو د پروژې خانې (Project Pane) څخه يوه تشه (Blank) پروژه خلاصه کړئ ، کوم چې په عامو حالاتو کې د غورني مينو (Menu bar) لاندې د وينډوز چپ طرف ته وي، . بيا Next، پروژه اوس تنظيم کولو ته ضرورت لري چې د سيميليټر (Simulator) په ذريعه پروګرام و چلوي

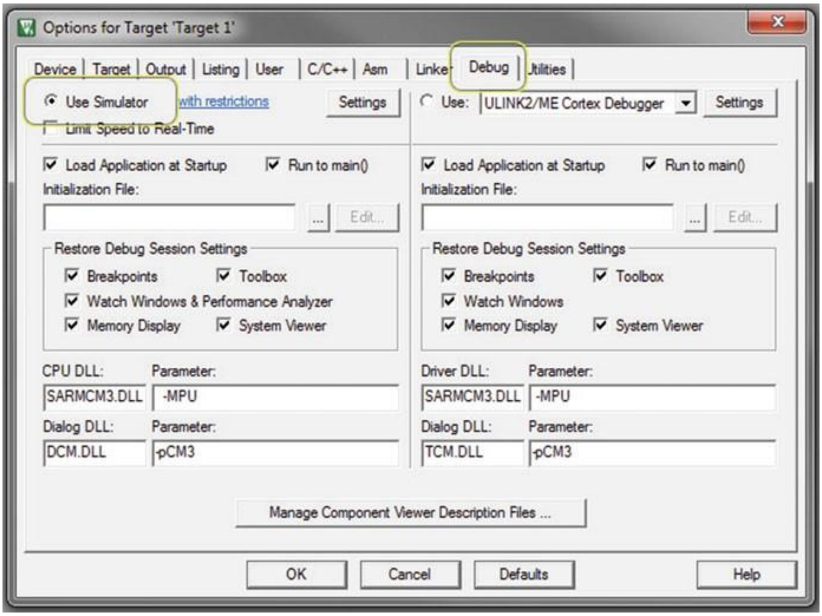


۴,۱۰ انځور: د پروژې تيظيمات تغيرول

- په *Target 1* فولډر باندی ښي کلیک وکړئ او *Options for Target "Target 1"..* . انتخاب کړئ (شکل: ۴,۱۰)
- په *Debug tab* کلیک وکړئ او *Use Simulator* انتخاب کړئ (شکل. ۵,۱۰)
- په *Target* کلیک وکړئ او *Use MicroLIB* انتخاب کړئ لکه څرنګه چې په شکل ۶,۱۰ کې دي
- ښی کلیک په *Source Group 1 folder* باندې د پروژې په پین کې دې وکړي او پروژې ته نوي شی *(Item)* ور اضافه کړئ لکه څرنګه چې په شکل. ۷,۱۰ دي.
- نوي اسمبلر منبع *(Assembler Source)* فایل جوړ کړئ *(Asm File* فایل یا *s.*)، ته یو نوم ورکړئ، او ثبت یي کړئ لکه څرنګه چې په شکل. ۸,۱۰ دی

اسمبلي پروګرام یو ټاکلي جوړښت لري او تعقیبوی یي چې لارښوونې ورته وایي. دا به وروسته په دی فصل کې تشریح شي، مګر دا اسمبلر ته لارښونه کوي چې څنګه د وروستي پروګرام (*Final Program*) ډیره برخه جوړه کړي. لکه زیات پورته لیول (*High-Level*) کمپیوټري ژبو پروګرامونه د اساسي میتود (*Main Method*) په دننه چلیږي لانديې مثال (*Routines Vectors*) او *(Reset_Handler)* استعمالوي او(کوم ته چې په عام حالت کې پروسیسر انتخاب شوی وي)

۱,۱۰ مثال: دا پروګرام دوه عدده لولي او دجمع حاصل یي په R3 کې ثبتوي (۹,۱۰ شکل)



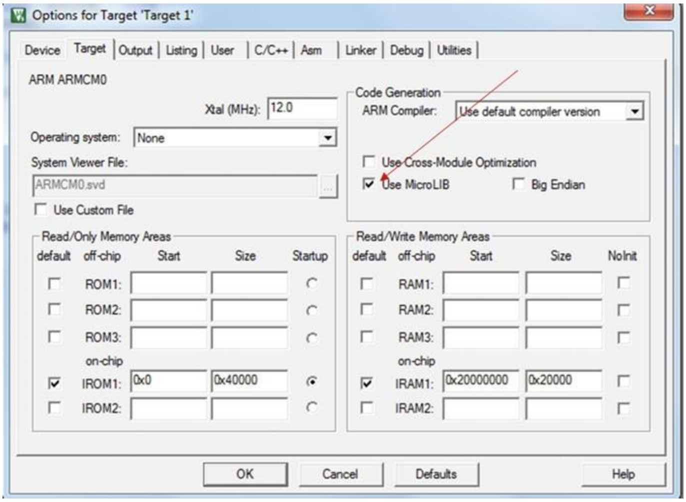
۵،۱۰ انځور: د سیمیلیټر (Simulator) د استعمالولو له پاره د پروژې تنظیم

۱،۲،۱۰ پروګرام بشپړونکی، د پروګرام راټولول (Assembling A Program):

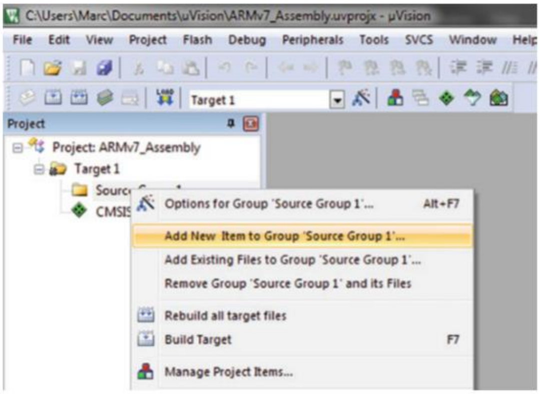
یوځل چې د سرچېنې یا سورس کوډ تیار شي بیا راټول (*Assembled*) شي، تاسو ممکن په سیمیلیټر (Simulator) کې د پروګرام ژباړلو او چلولو له پاره هڅه ولرئ. اولنی ګام د اسمبلي پروګرام جوړول دي د اسمبلر (Assembler) په کومک سره بیا د سرچېنې یا سورس کوډ ترجمه کړئ. دا کار هم له لیکمن نه کولی شي او هم یي د مینو له توکمو کولی شي دواړه اختیارونه لاندې لیست شوي مینو بار کې شتون لري په شکل. ۱۰،۱۰ ښودل شوي.

- F7 په واسطه ټول هدف (Target) شوي فایلونه چمتو (Build) کیږي (e.g., All Source Files In Target 1).
- Ctrl+F7 په اوسني فعال فایلونه چمتو (Build) کړئ (e.g., The file Being Modified).

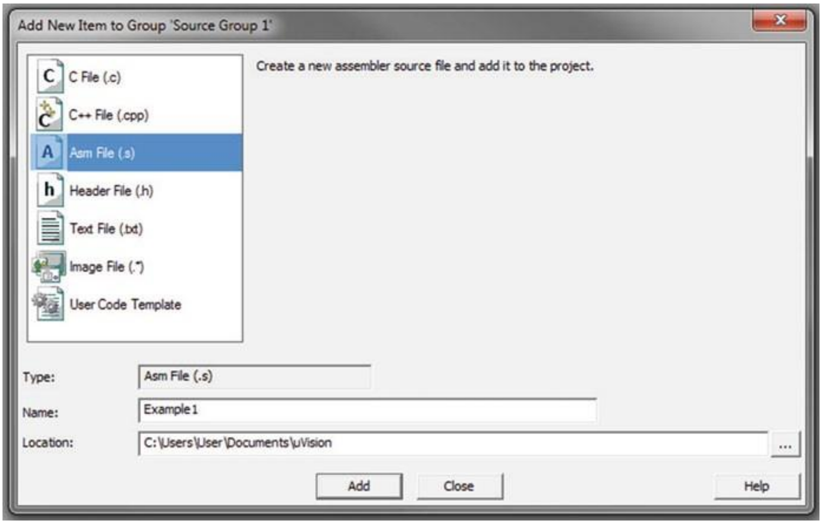
د وینډوز په ښکتنی برخه کې د راکړیز جوړښت برخه (Build Output Panel) غلطیانې ، خبرداري ، یا به شاید د پروژې کامیاب جوړیدل وښایي. په کامیابۍ سره چمتو شوي (Build) پروژه به داسې ښکاري لکه په شکل. ۱۱,۱۰ کې، که چیرې په کې نیمګړتیا وه دا به د غلطۍ وضاحت در کړي ترڅو پروګرام جوړونکي سره کومک وشي چې په کوم ځای کې کوډ غلطي لري.



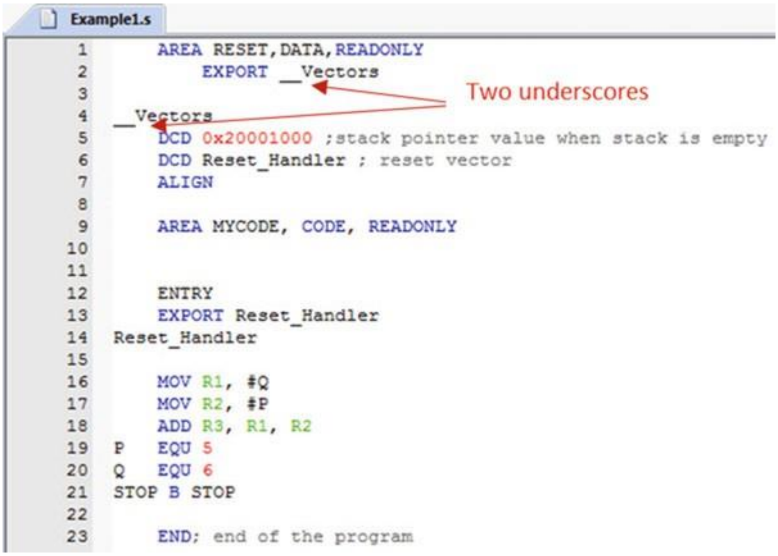
۶,۱۰ انځور: د تارګټ۱ (Target1) له پاره انتخاب (Micro LIB) استعمال کړئ



۷,۱۰ انځور: د یو شي (Item) اضافه کول پروژې ته

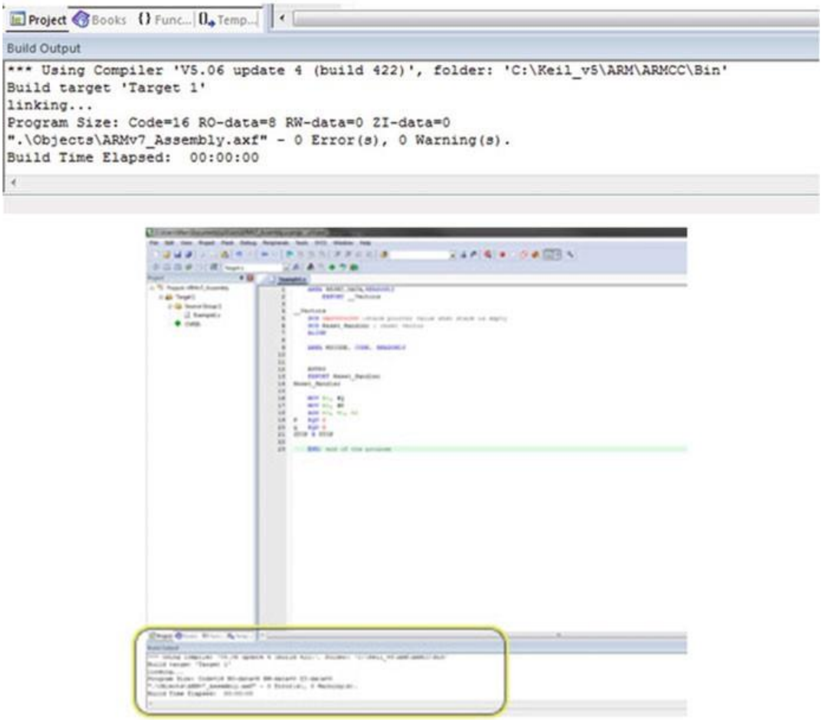


۸,۱۰انځور: د اسمبلي (Assembly) سرچېني يا سورس فايل جوړ کړئ



۹,۱۰ انځور: سرچېني يا سورس کوډ د ۱,۱۰ مثال له پاره

۱۰,۱۰ انځور: د اسمبلي پروګرام بټنې

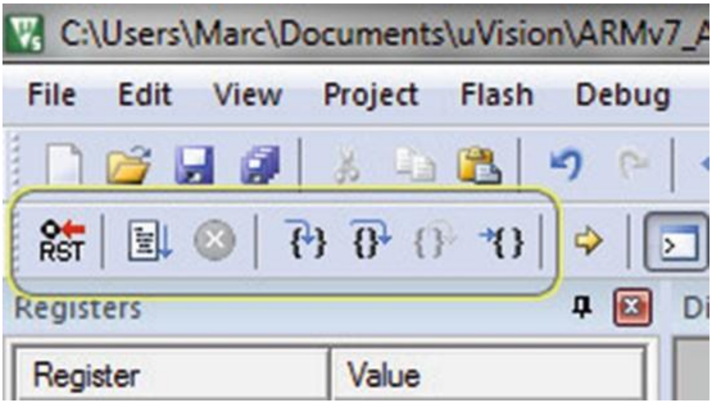


۱۱,۱۰ انځور: د بشپړ اسمبلي لپاره راکړیز (Output) جوړ کړئ

۲,۲,۱۰ د ډيبګر (Debugger)/ سيميليټر (Simulator) چلول: اوس چې تاسو د کوډ د يوې برخې کمپايل کول بشپړ کړل، تاسو غواړئ چې کوډ د ټسټ له پاره ډيبګ (خپل کوډ د موجوده غلطيو نه پاک) کړئ. د ډيبګ (Debug) شروع کولو له پاره، په مينو بار کې د Start/Stop Debug باندې کليک وکړئ ۱۲,۱۰ انځور.

µ*Vision* کېدای شي يو خبرداری ورکړي په ساده ډول د *Ok* تڼی کېکاږئ.

۱۲,۱۰ انځور: د ډيبګرشروع او ودرېدلStarting/Stopping



۱۳,۱۰ انځور: د ډيبګ د مينو د بټنو موقعيت

په عادي حالاتو کې راجستر (Registers) په چپ طرف د راجسټرونو په حصه کې ښکاريږي. که چيرې تاسو د *xPSR* لين سره نژدې په "+" کليک وکړئ، تاسو کولی شئ چې موجوده فلګ بيټ (*Flag Bits*) وګورئ دا د پروګرام د حالت راجسټر دی (PSR).

د *Disassembly* حصه موجوده معلومات ښکاره کوي، لکه د ماشين کوډ لارښووني او

شـاید دا وښـایي چې د کوم لین د سـرچېنې یا سـورس کوډ چلیدلی دی، او په کوم ادرس چلیدلی دی.

د پروګرام د چلولو له پاره تاسو شاید لاندې بټنې/ دلیکمن بټنی استعمال کړئ (۱۳,۱۰ انځور):

Reset - بیاځلې پروګرام چالانوی او CPU اولنی حالت ته بیایي

Run (F5) - بشپړ پروګرام چلوي تر هغې چې د قطع نقطه نه وي راغلې (که چېرته د قطع نقطه موجوده وي).

Stop - موجوده اجرا کېدونکې کوډ ودروي.

Step (F11) - د یوی دندی (Function) یو واحد ګام اجرا کوي

Step Over (F10) - د (Function) پوسیله یو مرحله اجرا کوي.

Step Out (Ctrl + F11) -موجوده دنـده (Function) ختموی او ودروی یي.

Run To Cursor Line (Ctrl+F10) - تر هغی پروګرام چلوي چې تر څو موجوده کرسر (Cursor) لین پوری نه وي رسیدلی.

۲,۱۰ مثال: د کوډ چلول ۸,۱۰ د انځور ته.

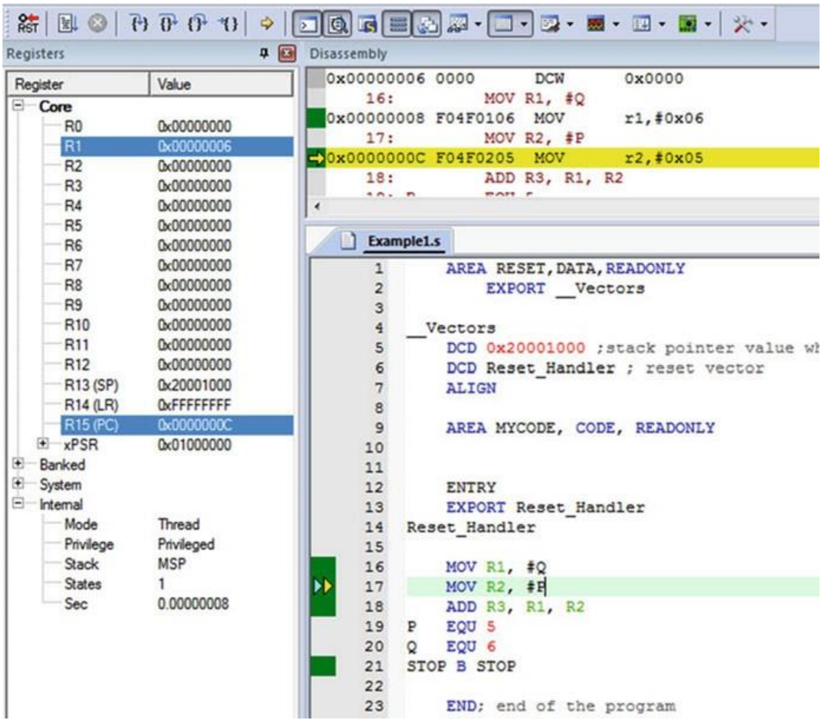
- اول، بیا ځلې کوډ ولیکې کټ مټ لکه په ۸,۱۰. کې چې ښکاري.
- د فایل ثبت کولو سره (یا Ctrl +S) کوډ بشـپړ کړی بیا د اسمبلي پرواګر ژبې څخه ماشـین کوډ ته د ترجمه کولو له پاره (یا F7) کېکاږئ. د چمتوشـوي حاصل (*Build Output*) حصه به داسې ښکاري لکه ۱۱,۱۰ انځور کې.

د د ډیبګ مینو لاندې Ctrl +F5 یا په کېکاږلو سمیلیټر/ډیبګر وچلوی سره. که چیري د ارزیابی طریقه (*EVALUATION MODE*) اشاره ښکاریده نو بیا OK

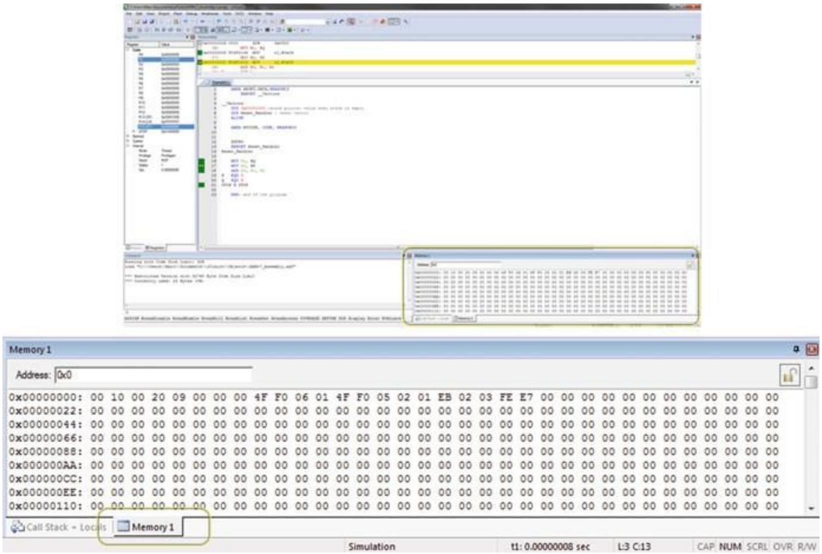
کېکارړئ.

- F11 یا کېکارړی د یو لین د قدم "Step" کولو له پاره او لاړښووني اجرا کړئ، او په هره حصه کې به تغيرات وګورئ. کله چې ډيبنګ کيږي، سمېليټر به موجوده چليدونکي يا تعديليدونکي معلومات په نښه کړئ. ۱۴,۱۰ انځور د مثال په توګه وګورئ کوم چې اولنی قدم "Step." څخه وروسته ستاسو سکرين بيا بشپړوي دا به زمونږ د کوډ اولنی بيټ اجرا کړي، MOV R1, #Q او په R1 راجستر کې Q (6) قيمت ثبتوي.

\- µVision's ډيبګر به سرچيني يا سورس کوډ ته راتلونکي لاړښووني نښاني کړي (MOV R2, #P),



۱۴,۱۰ انځور: ډېر وړ معلومات ډيبګر په اوتومات ډول نښاني کوي



۱۵,۱۰ انځور: د حافظې د وینډوز ځای په پنل ټب (Panel Tab) کې.

Disassembly حصه کې د ماشین کوډ او د لار ښوونو د حافظې ادرس سره د اصلي اسمبلي کوډ له نښې سره.

- د کوډ له لارې "Step" ته دوام ورکړئ ، هر ځل *F11* یا د Step بټنه کېکاږئ ترڅو پروګرام و چلول شي. په هر ګام او په هر راجستر کې بدلونونه وګورئ.

ستاسو د پروګرام د فعالیت د نظارت کولو بله لاره د اوسنۍ حافظې لیدل دي. ټول هغه لارښوونې چې د سرچینې یا سورس کوډ څخه ترجمه شوي او دمخه ټاکل شوي معلومات یې په حافظه کې ثبت شوي. *Disassembly* خانه به یوه لارښوونه په *32* بیټه شپاړلسو په قاعدې حافظه کې د هرې بلې لارښوونې سره جوخت ځرګنده کړې.

د مثال په توګه: د ډیبګ کولو پرمهال په حافظه کې د اوسني قیمت لیدو له پاره ، په ښي طرف کونج کې د حافظې ۱ حصه خلاصه کړئ (۱۵,۱۰ انځور). تر څلورو پورې وینډوز موجودې دي، او په عادې توګه د *Memory 1* فعاله وي. که چیرې نه وه نو بیا دا به مینو بار لاندې د View په کلیک کولو سره بیا Memory

❢ Windows بيا Memory 1 سـره فعالېږي. (يو فعاله وينډوز به دا نښـان په خوا کې لري).
په حافظه کې لټون، د شـپاړلسـو په قاعدې (Hexadecimal) سـره برابر ادرس به د حـافظې په برخه کې وليکي کوم چې تاسـو غواړئ و يي پلټئ کله نه چې دا بايټ ادرس ورکوونکې پروسيسـر (Byte-Addressable Processor) دی هر ادرس يو بايټ معلومات ساتي او دوه د ۱٦ په قاعده اعداد ښکاره کوي

۱٦،۱۰ انځور: په کېل پنځم ورژن کې د ARMCM3 پروسيسر له پاره د پروګرام خاکه.

```
    AREA RESET,DATA,READONLY
        EXPORT __Vectors

__Vectors
    DCD 0x20001000 ;stack pointer value when stack is empty
    DCD Reset_Handler ; reset vector
    ALIGN

    AREA MYCODE, CODE, READONLY


    ENTRY
    EXPORT Reset_Handler
Reset_Handler

   YOUR PROGRAM CODE HERE

   ; comments like this are ignored by the assembler
   ; mark them with a semicolon and use them to make
   ; notes throughout your programs


STOP B STOP

    END; end of the program
```

۳،۲،۱۰ د پـروګرام خـاکـه (Program Template): ٦،۱۰ انځور د ای ار ام (ARM) کـارټيکس ام ۳ (Cortex M3) له پاره د پروګرام ليکلو د کوډ نمونې خاکه په بر کې نيسي. دمنبع يا سورس کوډ د ريسيټ هينډلر (Reset_Handler) او سـټـاپ ليبلس (STOP label is) ترمنځ مو قيعت لري او کله چې پروګرام

وچليږی نو بيا دا اجرا کيږي. (ټول ليبلونه بايد خالي ځای ونه لري لکه خالي ځای او يا خانه (Tabs) چې په چپ طرف کې ده دا بايد حتمآ په کالم کې لومړی متن وي، او په حرف کې به لومړی لين کې وي).

۳,۱۰ د پروګرام ليکلو اصول (Programming Rules)

د کېسونو د اصولو (Case Rules): لارښوني، سمبولونه، او ليبلونه په دواړو غټو او وړو حرفو ليکل کېدای شي مګر يو ځای نه شي ليکل کېدای (مثال، MOV يا mov صحيح دی، مګر MoV يا moV صحيح نه دی)

ملاحظات (Comments): پروګرام ليکونکي کولی شي چې خپل ملاحظات له سيميکالن (;) څخه وروسته وليکي.

MOV R1, R2; Moving contents of R2 to R1

Base	Prefix	Example
Decimal (10)	-	17
Hexadecimal (16)	0x	0x11
Binary (2)	2_	2_00010001

۱۷,۱۰ جدول: د عددونو د ښرکندول چوکاټ

Label	Data
List	0x23
List+1	0x4A
List+2	0x56
List+3	0xF5

Memory Address	Data
0x400	0x23
0x401	0x4A
0x402	0x56
0x403	0xF5

۱۸,۱۰ جدول: د حافظې بلاک ډياګرام که چيري لست په 0x400 ادرس باندې ثبت وي

	Big Endian	Little Endian
0x400	0x20	0x0F
0x401	0x00	0x00
0x402	0x00	0x00
0x403	0x0F	0x20

۱۹,۱۰ جدول: د لوی انډيان او کوچنی انډيان مقايسه

۴,۱۰ د معلوماتو څرگندول او حافظه (Data Representation and Memory)

ای ار ام (ARM) پروسيسر يو لغات 4 بايټه او نيم لغات په 2 بايټه تعريفوي. معلومات د شپاړلسو، لسو او د دوو قاعدو په چوکاټ کې ښودل کيږي (۱۷. ۱۰ انځور). معلومات او کوډ په حافظه کې ساتل کيږي.۱۸,۱۰ انځور د حافظې بلاک ډياگرام څرگندوي. ادرس په حافظه کې د معلوماتو موقعيت څرگندوي. هر موقعيت د ای ار ام (ARM) پروسيسر حافظه کې يو بايټ ساتي. په اسمبلي پروگرام ليکلو ژبه کې ليبل لکه څنگه چې په ۱۴,۱۰ انځور کې ښودل شوي په حافظه کې د معلوماتو له پاره ادرس معلوموي. کله چې کوډ ترجمه شي اسمبلر په اتوماتيکه توگه تصميم نيسي چې کوم ادرس به د ليبل په کوم په ځای کې استعمال شي او يا يي بديل پيدا کوي چې د پروگرام په کوم بل ځای کې مناسب وي.

د دی لـه پـاره چې د حـافظې هر موقيعت يو بايټ سـاتي نو بايد پروگرام ليکونکي د کار کولو په وخت کې د يو لغات او نيم لغات خيال وسـاتي. پروسـيسر کولی شي چې له لوي او وړوکي انـډيـان (Endian) سـره کـار وکړي. ۱۹,۱۰ انځور0x2000000F لمبر څرگندوي کوم چې په دواړو سيستمونو کې په 0x400 کې ذخيره کيږي.

- د ای ار ام کـارټکس3 (ARM Cortex M3) د دی مثالو له پاره وړوکي انډيان (Endian) پروسيسر استعمالو.

۵,۱۰ لارښود (Directives)

لارښود یو اسمبلر کمانډ دی چې د اسمبلر له خوا اجرا کیږي. لارښووني هیڅکله ماشین کوډ نه تولیدوي. لارښووني د یوه کوډ شروع یا معلوماتو او د پروګرام پای مشخصوي. اي ان ډي (END)، یوه ساده لارښود دی چې د پروګرام پاي مشخصوي. بعضي مهم استعمالېدونکې لارښووني چې د ای ار ام (ARM) اسمبلر په ذریعه استعمالیږي په لاندي ډول دي:

- AREA- د حافظې یوه برخه تشریح کوي
- ENTRY- د پروګرام شروع تشریح کوي
- EQU- د لیبل د ثابت ورکولو (Assign) له پاره استعمالیږي

```
– X EQU 6
– MOV R1, #X ;  R1 = #6
```

۱,۵,۱۰ د معلوماتو لارښود (Data Directive): د معلوماتو لارښود د ډیټا ډولونه او اندازه مشخصوي:

- DCB(ثابت بایټ تشریح کوي)
- DCW (ثابت نیم لغت تشریح کوي)
- DCD (ثابت لغت تشریح کوي)
- SPACE (د حافظې په چوکاټ کې صفر په اندازه ځای نیسي)

DCB دا لارښود د لاندې یو یا زیاتو بایټونو (Bytes) په حافظه کې د ځای په ځای کولو له پاره استعمالیږي.

```
• list1 DCB 0xF, 10,2_00010001
  —list1 :  0x0F
  —list1 + 1: 0x0A
  —list1 + 2: 0x11
```

DCW دا لاربنود ثابت نيم لغت تشريح کوي (16 بيټونه، 2 بايټه) او د هر نيم لغت له پاره د حافظې دوه ځايونو ته اړتيا لري.

- $list1:\ 0x00-$
- $list1\ +1: 0xFF$
- $list1\ +2: 0x13$
- $list1\ +3: 0x00$

DCD دا د دېې له پاره استعمالېږي چې يو لغت (32 بيټه، 4 بايټه) تشـريح کړي او د هر لغت له پاره د حافظې دوه ځايونو ته اړتيا لري.

- $list1\ DCD\ 0x12345678, 0xFF$
- $list1: 0x78$
- $list1\ +1: 0x56$
- $list1\ +\ 2: 0x34$
- $list1\ +\ 3: 0x12$
- $list1\ +\ 4: 0xFF$
- $list1\ +\ 5: 0x00$
- $list1\ +\ 6: 0x00$
- $list1\ +\ 7: 0x00$

د حروفو سلسلې: د حروفو تسلسل ته جمله وايي. په ای ار ام اسمبلي کې جمله بايد حتماً باطله شي او په صفر ختمه شی کله چې تعريف شوي وي.

- $List1\ DCB\ “Assembly”, 0$
- $List2\ DCB\ “I\ have\ \$250”, 0$

اســمبلر يوه جمله په بايټونو (Bytes) تقســيموي او هر يوه برخه يي بيا په وړوکي انډيان (Little Endian) کې په ترتيب سره ثبتوي.

يو حرف: کله چې يو حرف په راجســتر يا حافظه کې ثبتيږی، حرف بايد حتما د نقل قول کولو (Single Quotation) په علامه کې ونيول شـي. بيا اسمبلر دا د شپاړلسو له قاعدې (Hexadecimal) اسکي کوډ (ASCII) ته واړوي.

- List DCB 'A'

فاصله: د حافظې هغه ځای چې د بیا وروسته استعمال له پاره ساتل (Reserves) شوي وي.

- List SPACE 20
- Reserves 20 bytes of memory starting at List.

۶،۱۰ حافظه په ویژن ۵ کې (Memory in µVision v5)

حافظه په ویژن (µVision) تقلید (Simulated) شـوی، او پروګرام کوشــش کوي چې حقیقی حالت بیان کړي. اکثره کوچني پروســیســرونه د رام (Read-(ROM) Only Memory) او رم (RAM) ترکیب اسـتعمالوي او رام په داسـې حالت کې چې برق لاړ شـي هم یو پروګرام ځان سره ساتي اویا که کمپیوټر بند هم شی او بیرته چالان شي نو بیا هم پروګرام له ځآن سـره سـاتي مګر رام (ROM) په داسـې حالاتو کې بیا پروګرام ځان سره نشي ساتلی.

د دی چارې ترسره کولو له پاره حافظه هیڅ یا بعضې یا لاندې ټولي اجازی لري.

Exec- حافظه لارښوونې ځان سره لري او کېدای شي چې پروګرام اجراشي.

Read- حافظه لوستل کېدای شي.

Write- حافظې ته لیکل (*Write*) کېدای شي.

پـه عـامو حـالاتو کې، د حـافظې پـه هغه ځای کې زیاتې لارښـــوونې او معلومات یواځې په دواړو د اجرا (Exec) او لوستل کېدو (Read) له پاره په نښې سره زېرمه کوي، نه د لیکلو (Write) له پاره. دغه میمک (Mimics) رام کله چې لارښوونې په کې زېرمه شـي هغه بیا هم لوستل کېدای شي او هم چلیدی شي مګر دغو ځایونو ته د لیکلو هڅه به کار ور نه کړي. اسمبلر هم نشي کولی چې د دې نتیجې له پاره کوشش وکړي.

۳،۱۰ مثال: د List1 محتویات وروسته د اجراکېدلو به څه وي، کله چې په رام (*ROM*)

کې ذخیره وي؟

```
ADR R0, LIST1 ; Store The Address Of List 1 In R9
MOV R1, #2 ; Store 2 in R1
LDRB R2, [R0]; Load The Byte At List 1 Into R2
ADD R3, R1, R2; R3←R1 + R2 (0 + 2 = 2)
STRB R3, [R0] ; Store R3 (#2) At List 1 List 1 =0x00
```

دا چې ویژن (μVision) او ای ار ام (ARM) منبع یا سورس کتابتونونه پروګرام ته دا وړتیا ورکوي چې په اسانۍ سره د رام (ROM) په ځای په رام (RAM) کې زېرمه شــوی معلومـات ځـای په ځای کړي، دا موضــوعات د دی څپرکي مربوط نه دي.په هر صـورت چې کوم ادرس د لوسـتلو (*Read*) او لیکلو(*Write*) له پاره خاص شوی بیا هم تاسو رام (RAM) په حافظه کې د معلوماتو د زېرمه کولو له پاره استعمالولی شئ.

پـه عامو حالاتو کې، د ARMCM3 پروســیســر، د حافظې موقعیتونه – 0x20000000 0x20020000 د لوستلو(*Read*) او لیکلو (*Write*) له پاره په رام (RAM) کې په ویژن (μVision) کې مشخصوي.

۴،۱۰ مثـال: د لانـدې لارښــوونو د اجرا کېـدو څخـه وروســتـه دحـافظې محتویـات به 0x200000000 څه وي؟

```
List1 DCB 0x0
ADR R0, List1 ; Store The Address of List1 in R0
MOV R1, #2 ; Store 2 In R1
LDRB R2, [R0]; Load The Byte At List 1 Into R2
ADD R3, R1, R2; R3←R1 + R2 (0 + 2 = 2)
MOV R4, 0x20000000; Store Our Initial RAM Address In R4
STRB R3, [R4]; Store R3 (#2) At 0x20000000;
List1 = 0x00
0x20000000 0x02
```

ففبب (ffcC)

- د یو پروسیسر سمولیشن (Simulation) چې په مختلفو پروسیسرونو باندې چلیږی د کراس اسمبلر (Cross-Assembler) په نوم یادیږي
- د ای ار ام (ARM) پروسیسر له پاره مختلف پروګرامونه موجود دي لکه Keil, IAR, او GNU.
- د ای ار ام (ARM) لارښوونې او لیبلونه هم په وړو او هم په غټو حروفو لیکل کیږي مګر یو ځایي نه لیکل کیږي.
- د ای ار ام (ARM) پروسیسر یو لغت په 4 بایټونو، نیم لغت په دوه بایټونو سره څرګندوي، او دا د دوه یزو (Binary), (2_1000011) د شپاړلسو په قاعدې (0x24), او د لسو په قاعدې (Decimal) 45 سره ښودل کیږي.
- د معلوماتو لارښووني دا دي

DCB (ثابت بایټ تشریح کوي) ,DCW (ثابت نیم لغت تشریح کوي) ,DCD (ثابت لغت تشریح کوي SPACE ,(د حافظې په چوکاټ کې صفر په اندازه ځای نیسي)

- د حافظې په موقعیت کې یو بایټ ښودلو له پاره DCB استعمالیږي لکه:

LIST DCB 0x32

- د حافظې په موقعیت کې نیم لغت ښودلو له پاره DCW استعمالیږي لکه:

LIST DCW 0x3245

- د حافظې په موقعیت کې یو لغت ښودلو له پاره DCD استعمالیږي لکه

LIST DCD 0x87673245

- د فاصلی لارښود د حافظې د موقعیت د ریزرف کولو له پاره د صفر قیمت په ورکولو سره په ټولو موقعیتونو کی استعمالیږي لکه:

LIST SPACE 20

- د حروفو جملې باید په (0) سره ختمې شي لکه:

List DCB “WELCOME”,0

- يو حرف د حـافظې په موقعيت کې بايد حتما يو نقل قول (Single Quotations) په علامه کې ونيول شي لکه:

List DB ‘A’

ستونزې

۱. یو پروګرام ولیکې چې د List 1 عناصر د List 2 د عناصرو سره جمع کړئ.

```
List1        DCB 0x23, 0x45, 0X23,  0x11
List2        DCB 0x0
```

۲. یو پروګرام ولیکئ چې لوي عـدد پیـدا کړئ او هغه د حافظې د List 3 په موقعیت کې زېرمه کړي.

```
List1   DCD   0x23456754
List2   DCD   0X34555555
List3   DCD   0x0
```

۳. یو پروګرام ولیکي چې د معلوماتو مجموعه د حافظې د LIST په موقعیت کې پیدا کړي اود حلقې (Loop) په کارولو سره د هغې مجوعه د حافظې په خاص موقعیت کې زېرمه کړئ.

```
List     DCB    0x23,    0x24,   0x67,   0x22,  0x99
SUM    DCB   0x0
```

۴. د لانـدې پروګرام لـه اجراکېـدو څخه وروســتـه د R5 په ذریعه د R1 راجســتر محتویات وښایي.

```
AREA NAME,          CODE,READONLY
        EXPORT         SystemInit
        EXPORT__main
        ENTRY
        SystemInit
        __main
AREA Directives,      CODE,READONLY
        ENTRY
        ADR   R0,      LIST1
```

```
        LDRB R1,    [R0]
        LDRB R2,    [R0,    #1]!
        LDRB R3,    [R0,#1]!
        LDRB R4,    [R0,#1]!
        LDRB R5,    [R0,#1]
List    DCB  0x23,  0x24,  0x67,  0x22,  0x99
        align
        END
```

۵. د اسمبلي پروګرام ژبې داسې یو پروګرام ولیکئ چې د 5, 3, 0 او R12 کې لږ مهم موقعیت تش کړي او نور بیټونه (Bits) باید تغیر ونه کړي (ای ار ام (ARM) لارښوونې و کاروئ).

حل:

```
LDR   R1,    #0x00000069
BIC   R12,   R12,   R1
```

۶. لاندې پورته لیول ژبې (HLL) پروګرام اسمبلي ژبې ته یو پروګرام ولیکئ.

```
IF   R1= R0
Then
ADD  R3,  R0,  #5
Else
SUB  R3,  R0,  #5
```

7. یو پروګرام ولیکئ چې LIST1 او LIST2 د حافظې موقعیتونه ولولي او مجموعه یې په LIST3 کې زېرمه کړي.

```
LIST1 DCD   0x00002345
LIST2 DCD   0X00011111
LIST3 DCD    0x0
```

8. د فرعی تګلاری (Subroutine) په کارولو سره یو پروګرام ولیکئ چې دوه عددونه سره ضرب کړي.

۹. یو پروګرام ولیکې چې د غیر مستقیم ادرس سیستم له لارې اته عددونه سره جمع کړي.

10. يو پرواگرام وليکئ چې د وروسته نه ترتيب شوی (Post-Index) په استعمال سره اته عددونه سره جمع کړي.

LIST DCB 0x5, 0x2,0x6,0x7 ,0x9,0x1,0x2,0x08

11. يو پروگرام وليکئ چې لاندې پورته ليول ژبې (HLL) لارښوونې ای ار ام (ARM) لارښوونو ته تبديلی کړي.

```
IF   R1=R2   AND   R3>R4   then
R1= R1 +1
Else
R3=R3 +R3*8
Endif
```

۱۲. د لاندې پروگرام له اجراکېدو ورسته به د R4 محتويات څه وي.

```
__main
LDR    R1, =0xFF00FF
ADR    R0,    LIST1
LDR    R2,    [R0]
AND   R4,     R2,     R1
LIST1  DCD   0X45073487
```

13. يو پروگرام وليکې چې لاندې پورته ليول ژبې (HLL) لارښوونې ای ار ام (ARM) لارښوونو ته تبديلی کړي.

```
If   R1=R2    then
R3= R3+1
IF    R1<R2   Then
R3=R3-1
If    R1>R2     Then
R3=R3-5
```

۱۴. د Y د محاسبه کولو له پاره یو فرعی تګلاره (Subroutine) ولیکئ چې X + X*2 Y=+ 5 محاسبه کړی، فرض کړئ چې X ښودل کیږي په

```
LIST   DCB    0x5
LIST1  DCB    0x5
```

۱۵. یو پروګرام ولیکئ چې 16 ځله R1 وخرخوي فرض کړئ چې R1 مساوي دی له 0x12345678.

16. یو پروګرام ولیکې چې دوه عددونه مقایسه کړئ او لوي عدد د List د حافظې په موقعت کې ذخیره کړئ.

```
M1              EQU 5
N1              EQU 6
LIST2  DCB 0x0
```

۱۷. یو پروګرام ولیکئ چې یو لغت د حافظې List و لولي د R5 راجستر B7 په ذریعه B4 بیټ ځای تش (Clear) کړی، فرض کړي R5 مساوي کیږي له 0xfffffff سره.

```
LDR     R0, =  0x000000F0
LDR    R5, =   0xFFFFFFFF
```

۱۸. یو پروګرام ولیکئ چې راجستر R1, R2, R3, اوR4 د LIST د حافظې د موقعیت څخه چالان کړي.

```
LIST DCD 0x12345aaa, 0x0000bbbb, 0x0000cccc,
0X00000DDD
```

ضمایم: ډیجیټل ډیزاین لابراتواري تجربې د LOGISIM په استعمال سره

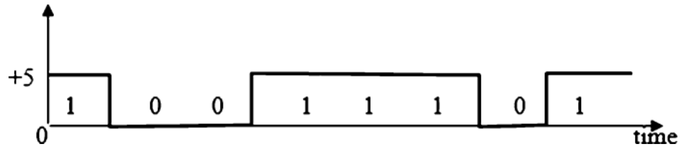
اول څپرکی

پوښتني او ځوابونه

۱. دري کمپیوټري راکړیزی آلې لیست کړئ.

مانیتور، لاسپیکر، پرنټر

۲. ډیجیټل سګنل مثال یو ولیکئ.



(a) بایټ = 8 بیټه

(b) نیم لغت = 16 بیټه

(c) لغت = 32 بیټه

۳. لاندې د لسو په قاعدې اعداد د دوو قاعدې اعدادوته تبدیل کړئ:

(a) 35=1000011

(b) 85 =1010101

(c) 23.25

$0.25 * 2 = 0.5\ A_{-1} = 0$

$0.5 * 2 = 1.0\ A_{-2} = 1$

$23.25 = 10111.01$

۴. لاندې د دوو په قاعدې اعداد شپاړلسو قاعدې اعدادو ته تبدیل کړئ:

(a) 1110011010 = 39 A
(b) 1000100111 = 227
(c) 101111.101 = 101111.1010 = 2f.A

۵. د لاندې ډیجیټل سـګنل فریکونسـي درکړل شـوي، د ډیجیټل سـګنل د سـاعت دوران(Clock Cycle) پیدا کړئ.

(a) 10 Hz T = 1/F = 1/10 = 0.1 Second
(b) 200Hz T = 1/F = 1/200 = 0.005 Second
(c) 10000 Hz T=1/10000 = 0.1 Milliseconds
(d) 4 Mhz $T = 1/ 4 *10^6 = 0.25 *10^{-6}$ = 0.25 Microseconds

۶. لاندې د لسو په قاعدې اعداد د شپاړلسوقاعدې ته تبدیل کړئ:

(a) $(234)_{10}$ =
Quotient Remainder
234/16 =14 10 = A In Hex = A0
14/16 =0 14 = E in hex = a1
$(a_1\ a_0)_{16} = (EA)_{16}$

(b) $(75)_{10}$
Quotient Remainder
75/16 4 11 = $(B)_{16}$
4/16 0 4= $(4)_{16}$
(c) $(75)_{10} = (4B)_{16}$

۷. د لاندې اعدادو دوه ییز کامپلمنټ (Two’s Complements) پیدا کړئ:

1. 11111111
Two’s Complement Of A Number = Complement Of The Number +1
11111111 = 00000000 +1 = 00000001
2. 10110000 = 01001111 +1 = 01010000
3. 10000000 = 01111111 +1 = 10000000
4. 00000000 = 1111111 1+1 = 0000000

۸. $(465)_{10}$ د Bcd په واسطه وښایاست؟

(0100 0110 0101) BCD

۹. (110010000100) BCD د لسو قاعدې په واسطه وښایاست؟

Invalid Number

۱۰. لانـدې بی علامي اعـداد د دوه ییز کـامپلمنـټ (Two’s Complement) پـه استعمال سره منفي کړئ

(a) 11110011 – 11000011

Add Minuend To The Two’s Complement Of Subtrahend.

1. Two’s Complement Of 11000011 = 00111100+1=00111101
2. Add The Result Of Step 1 To The Minuend.

```
 11110011
+00111101
---------------
```

100110000 - Adding The Two 8 Bits Results In 9 Bits, The Most Significant Called Carry; Discard The Carry, And The Result Is Positive + 00110000.

(b) 10001101 – 11111000

Two’s Complement Of 11111000
=00000111+1=00001000

```
 10001101
+00001000
---------------
```

10010101 – Adding Two 8 Bits Results In 8 Bits; Take Two’s Complement Of Sum, And The Result Is Negative.
Two’s complement of 0010101=01101010+1=01101011

Result – 01101011

۱۱. کوم یو د 16 بیټه دوه ییز لوي قیمت لرونکي اعداد دي چې ښودل کیږي په

(a) Unsigned Number

(b) Signed Magnitude

(A) $2^{16} - 1$

(B) + 2^{15}—1 Or —2^{15}—1

۱۲. لاندي IEEE 745 سیګنل دقت لرونکي تبدیل کړی د لسو قاعدې اعدادو ته:

(a) 1 10000100 01110000000000000000000

(b) 0 01111100 11100000000000000000000

(A) S = 1 Mean Number Is Negative

Exponent = 10000100 – 01111111 = 132—127 = 5

Mantissa = 1.0111

Number = —1.011 1* 2^5 = —1011100 = —92

(B) S=0 Mean Number Is Positive

Exponent = 01111100 – 01111111 = 124 – 127 = —3 = —11

Mantissa = 1. 111

+ 1.111 * 2^{-3} = 0.001111 = 0.234375

۱۳. لاندې هر یو علامه لرونکي میګنیچوډ (Signed Magnitude) اعداد د لســو قاعدې اعدادو ته تبدیل کړئ:

(a) 11000011

(b) 10001111

(A) The Most Significant Bit Represents The Sign

-1000011= —35

(B) –00001111 = —15

۱۴. لاندي د جمعی عملیه ترسره کړئ:

$(0f4a)_{16} + (420b)_{16} + (5155)_{16}$

دوهم څپرکی

ځوابونه

۱. که چېري A=11001011 او B=10101110 ، د لاندې معادلو نتیجه به څه وي؟ د لاندې علامو په استعمال سره ؟

(a) A And B
(b) A Or B

(a) Performing Bit By Bit And Operation Of
A =11001011
B = 10101110
A And B = 10001010

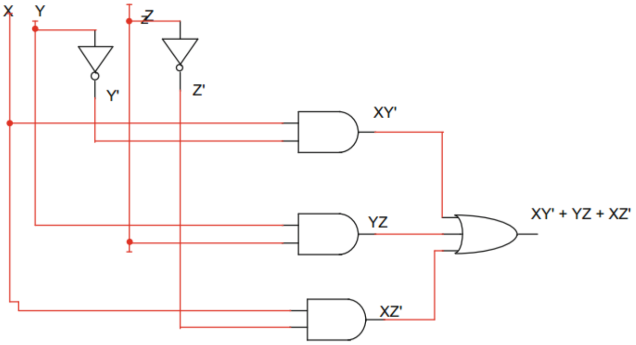
(b) Performing Bit By Bit Or Operation A = 11001011
B = 10101110
A Or B = 11101111

2. د لاندې تابع ګانو له پاره منطقي سرکټ رسم کړئ:

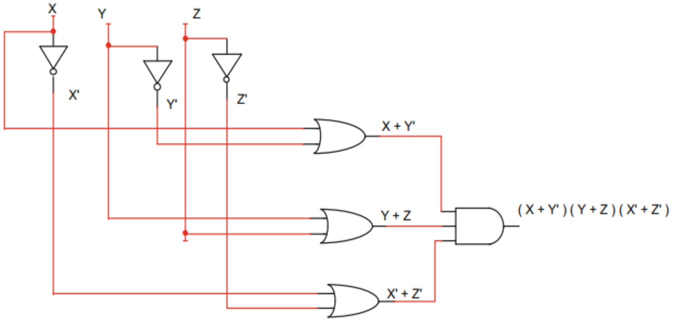
(a) $F(X,Y,Z) = Xy' + Yz + Xz'$

(b) $F(X,Y,Z) = (X + Y') (Y + Z) (X' + Z')$

a)



b)



3. لاندي تابع گانی ساده کړئ:

(a) $F(X, Y, Z) = Xy + X'y + Xz$

حل

$F(X, Y, Z) = Y(X + X') + Xz = Y + Xz$

(b) $F(X, Y, Z) = (X + Y)(X' + Y + Z)$

حل

$F(X, Y, Z) = (XX' + XY + XZ + X'Y + YY + YZ)$ Where $XX' = 0$

And $YY = Y$ $F(X, Y, Z) = (XY + XZ + X'Y + Y + YZ)$

$F(X, Y, Z) = Y(X + X' + 1 + Z) + XZ$ $F(X, Y, Z) = Y + XZ$

$F(X, Y, Z) = XY'Z + XYZ + Y'Z$ $F(X, Y, Z) = XZ(Y' + Y) + Y'Z$

$F(X, Y, Z) = XZ + Y'Z$

(c) $F(X, Y, Z) = Xy + Yx'z$
$F(X, Y, Z) = Y(X + X'z)$ Where $X + X'z = X + Z$ $F(X, Y, Z) = Y(X + Z)$

(d) $F(X, Y, Z) = X'y + Yxz'$
$F(X, Y, Z) = Y(X' + Xz') = Y(X' + Z')$

(e) $F(X, Y, Z) = Xy + (X + Y + Z)'x + Yz$
$F(X, Y, Z) = Xy + (X'y'z')X + Yz$ $F(X, Y, Z) = Xy + Yz$

(f) $F(X, Y, Z) = (Xy)' + (X' + Y + Z')'$

(g) $F(X, Y, Z) = X' + Y' + Xy'z$
$F(X, Y, Z) = Y'(1 + Xz) + X' = X' + Y'$

۴. که چيري A=10110110 او B=10110011 نو بيا پيداکړئ

(a) A Nand B
(b) A Nor B
(c) A Xor B

(a) NAND Each Bit Of A With Corresponding Bit Of B
A = 10110110
B = 10110011
A Nand B = 01001101

(b)

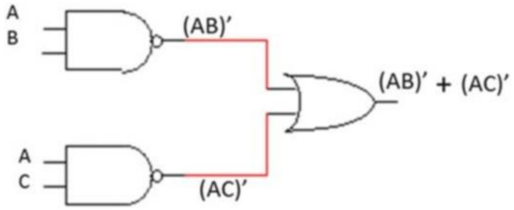
A = 10110110
B = 10110011
A Nor B = 01001000

(c)

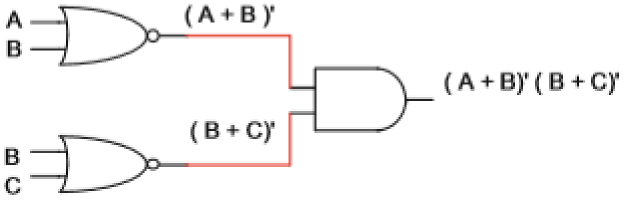
A = 10110110
B = 10110011
A Xor B = 00000101

۵. د لاندې منطقي سرکټ راکړيز ښکاره کړئ:

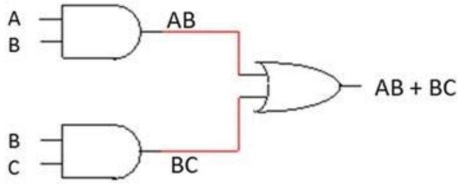
a)



b)



c)



۶. د لاندې منطقي سرکټ راکړيز تابع پيدا کړئ:

7. د لاندې هر يو تابع له پاره صدق جدول (Truth Table) جوړ کړئ:

(a) $F(X,Y,Z) = Xy' + Xz' + Yz$

(b) $F(X,Y,Z) = (X + Y)\ (X + Z')$

(c) $F(X,Y,Z) = Xy\ (Y + Z')$

X	Y	Z	Y′	(XY)	(XY)′	(X+Y′)	X (X+Y′)	(XY)′+X (X+Y′)
0	0	0	1	0	1	1	0	1
0	0	1	1	0	1	1	0	1
0	1	0	0	0	1	0	0	0
0	1	1	0	0	1	0	0	0
1	0	0	1	0	1	1	1	1
1	0	1	1	0	1	1	1	1
1	1	0	0	1	0	1	1	1
1	1	1	0	1	0	1	1	1

$$F(X,Y,Z) = (X + Y + Z')' (X' + Y')$$

X	Y	Z	X′	Y′	Z′	(X′+Y′)	(X+Y+Z′)	(X+Y+Z′)′	(X+Y+Z′)′(X′+Y′)
0	0	0	1	1	1	1	1	0	0
0	0	1	1	1	0	1	0	1	1
0	1	0	1	0	1	1	1	0	0
0	1	1	1	0	0	1	1	0	0
1	0	0	0	1	1	1	1	0	0
1	0	1	0	1	0	1	1	0	0
1	1	0	0	0	1	0	1	0	0
1	1	1	0	0	0	0	1	0	0

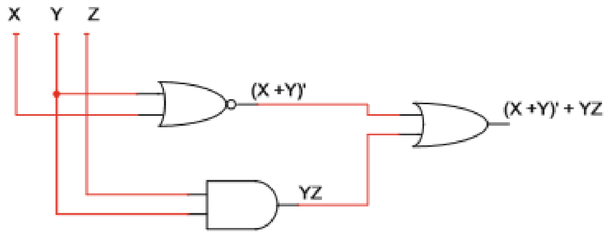
X	Y	Z	X′	Y′	(X′+Y′+Z)	(X+Y)	(X′+ Y′+Z) (X+Y)
0	0	0	1	1	1	0	0
0	0	1	1	1	1	0	0
0	1	0	1	0	1	1	1
0	1	1	1	0	1	1	1
1	0	0	0	1	1	1	1
1	0	1	0	1	1	1	1
1	1	0	0	0	0	1	0
1	1	1	0	0	1	1	1

8. د لاندي تابع گانو له پاره منطقي سرکټ رسم کړئ.

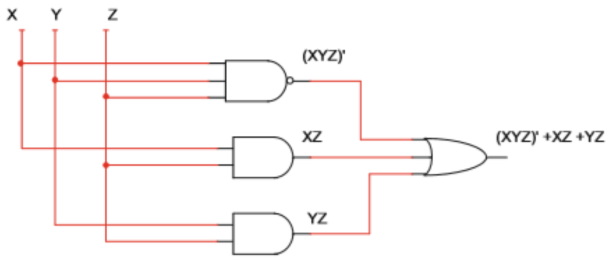
(a) $F(X,Y,Z) = (X + Y)' + Yz$

(b) $F(X,Y,Z) = (Xyz)' + Xz + Yz$

a)



b)



دریم څپرکی: حل

مشکلات

۱. د لاندې تابع گانو له پاره د صدق جدول (Truth Tables) جوړ کړئ:

(A) $F(X,Y,Z) = \sum(1,3,6,7)$

X	Y	Z	F
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

(B) $F(X,Y,Z) = \Pi(1,3,4)$

مسکټرم په جدول کې د صفر پواسطه ښودل کیږي

X	Y	Z	F
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

(C) $F(W,X,Y,Z) = \sum(1,4,7,10,12,15)$

منټرم په جدول کې د یو پواسطه ښودل کیږي

(D) $F(W,X,Y,Z) = \Pi(2,3,4,7,10,11,12,13)$

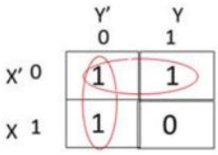
مکسټرم په جدول کې د صفر په واسطه ښودل کیږي

$F(X,Y,Z) = Y'z' + Yz$

۲. لاندې تابع ګانی د کارناف نقشې (K-Map) په استعمال سره ساده کړئ:

a)

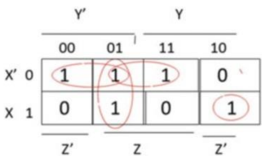
X	Y	F
0	0	1
0	1	1
1	0	1
1	1	0



$F(X,Y) = X' + Y'$

b)

X	Y	Z	F
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0



$F(X,Y,Z)$ ¼ $X'y' + X'z + Y'z + Xyz'$

c)

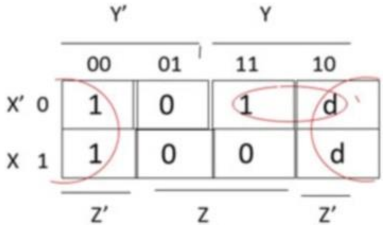
A	B	C	D	F
0	0	0	0	1
0	0	0	1	0
0	0	1	0	1
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

$F(W,X,Y,Z) = Wz + W'x'y' + W'xz'$ Sop

۳. لاندي تابع ګانی ساده کړئ په داسې حال کې چې D د نه پاملرنې (*Don't Care*) تابع ده:

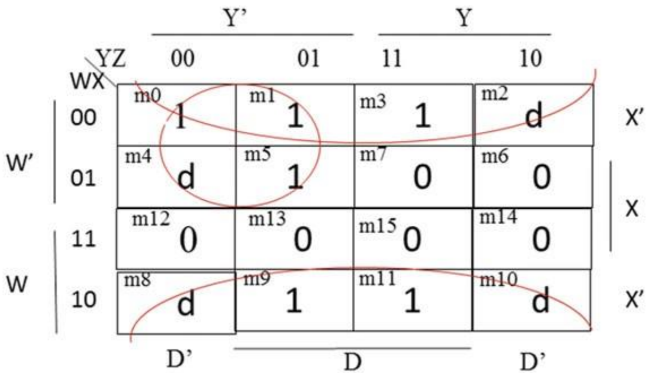
$F(X,Y,Z) = \sum(0, 3, 4)$

$D(X,Y,Z) = \sum(2, 6)$



$F(X,Y,Z) = Z' + X'y$

$F(W,X,Y,Z) = \sum(0, 1, 3, 5, 9, 11)$



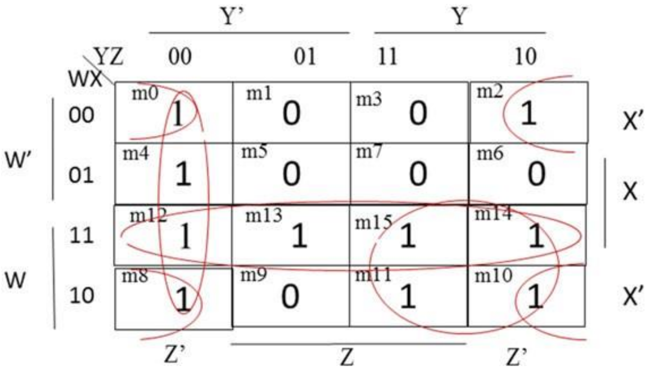
$D(W,X,Y,Z) = \sum(2, 4, 8, 10)$

۴. لاندې تابع ګانی ساده کړئ او منطقي سرکټ ورته رسم کړئ د لاندې ګیټونو په استعمال سره

(A) NAND Gates

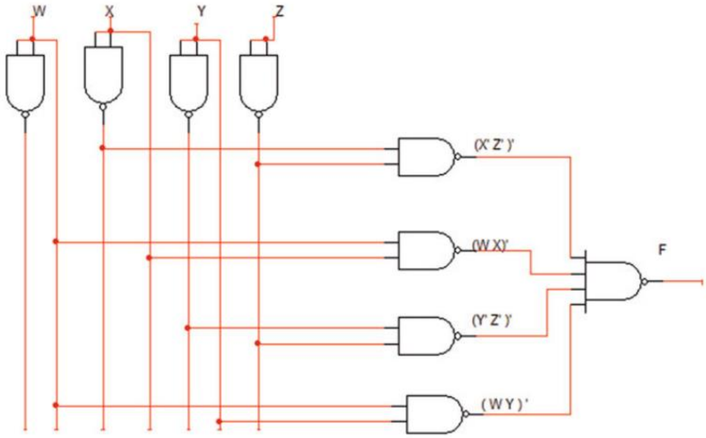
(B) NOR Gates

$Wy'x'z' + Wy + Wx + Xy'z' + W'x'z' = F(W,X,Y,Z)$



$F(W,X,Y,Z) = X'z' + Wx + Y'z' + Wy$ $F(W,X,Y,Z) = [(X'z' + Wx + Y'z' + Wy)']'$

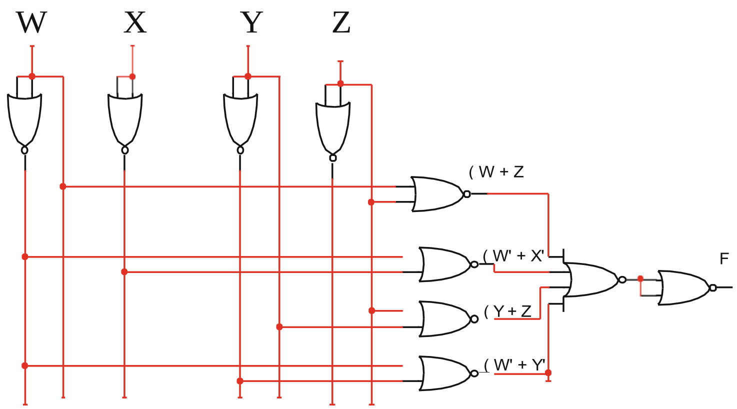
$F(W,X,Y,Z) = [(X'z')' (Wx)' (Y'z')' (Wy)']'$ Nand Form



$F(W,X,Y,Z) = [(X'z')' (Wx)' (Y'z')' (Wy)']'$

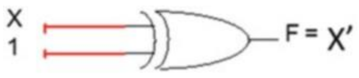
$F(W,X,Y,Z) = [(X + Z) (W' + X') (Y + Z) (W' + Y')]'$

$F(W,X,Y,Z) = (X + Z)' + (W' + X')' + (Y + Z)' + (W' + Y')'$ Nor Form



څلورم څپرکی

۱. د لاندې ګیټونو له پاره راکړېز (Output) پیدا کړئ:



2. یو منطقي سـرکټ د درېو ورکړېز او یو راکړېز سـره ډیزاین کړئ، په داسـې حال کې چې ورکړېز پیریټي بیټ (Parity Bit) د راکړېز هم جوړوي، فرض کړئ چې کله صـفر هم وي

(a) Show The Truth Table.
(b) Find Output Function.
(c) Draw Logic Circuit.

X	Y	Z	F
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

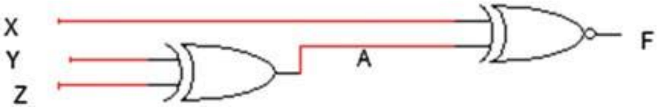
M6+ M5 + M3 + M0 = F(X,Y,Z)

Y' Y

	00	01	11	10
X' 0	1	0	1	0
X 1	0	1	0	1

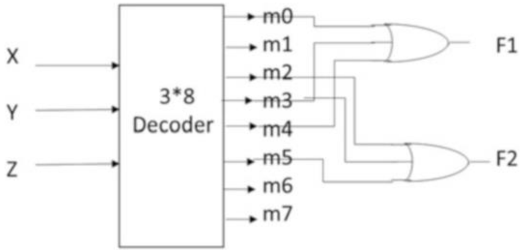
Z' Z Z'

$F(X,Y,Z)=X'y'z'+X'yz+Xy'z+Xyz'=X' (Y'z'+Yz)+X (Y'z+Yz')$
If $Y'Z + YZ'=A$ Then $Y'Z'+YZ=A'$
Therefore Function F Can Be Written As $F(X,Y,Z) = X'a' + Xa$
= X Xnor A

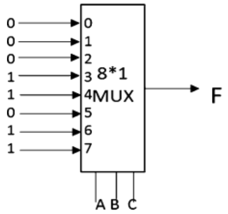


۳. يواځې د يوه اصـلي حالت ته تبديلوونكى (Decoder) او يوه بيرونى گيټ (Gates) په استعمالولو سره لاندي تابع گاني تطبيق کړئ:

$F1(X,Y,Z) = \sum(0, 3, 4)$
$F2(X,Y,Z) = \sum(2, 3, 5)$
The Function Requires A 3*8 Decoder.

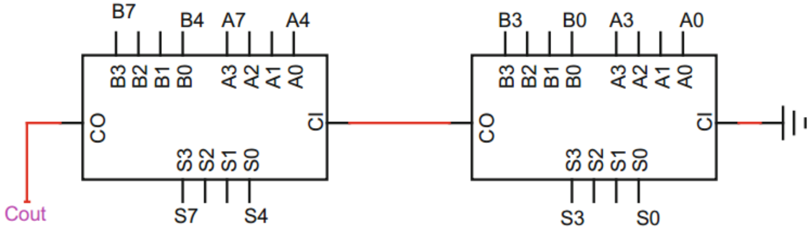


۴. لاندې ملټيپلکسرونه درکړل شوی، د دوی له پاره چوکاټ (Table) تکمیل کړې.



A	B	C	F
0	0	0	0
0	1	1	1
0	1	0	0
1	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
0	0	1	0

۵. د 4 بیټه دوه یز جمع کوونکي په استعمال سره اته بیټه دوه یز جمع کوونکی ډیزاین کړئ.

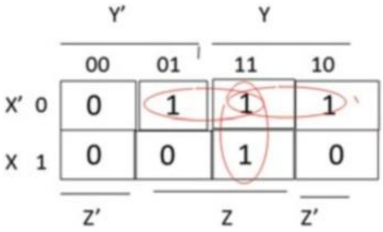


6. د دري ورکړيز او دري راکړيز سره یو ترکیبي منطق ډیزاین کړئ، که چیري ورکړيز 0, 1, 2 یا 3 وي نو بیا راکړيز 3 لوي دی له ورکړيز څخه، که چیري ورکړيز 4, 5, 6 یا 7 وي نو بیا راکړيز 3 وړوکی دی له ورکړيز څخه.

X	Y	Z	A	B	C
0	0	0	0	1	1
0	0	1	1	0	0
0	1	0	1	0	1
0	1	1	1	1	0
1	0	0	0	0	1
1	0	1	0	1	0
1	1	0	0	1	1
1	1	1	1	0	0

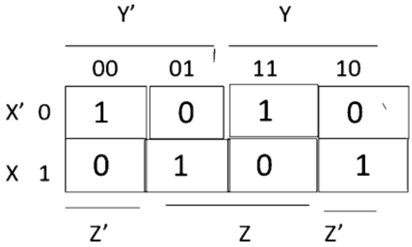
Making K-Map For A, B, And C

K-Map For A



A (X,Y,Z) $X'z$ $X'y$ Yz

K-Map For B



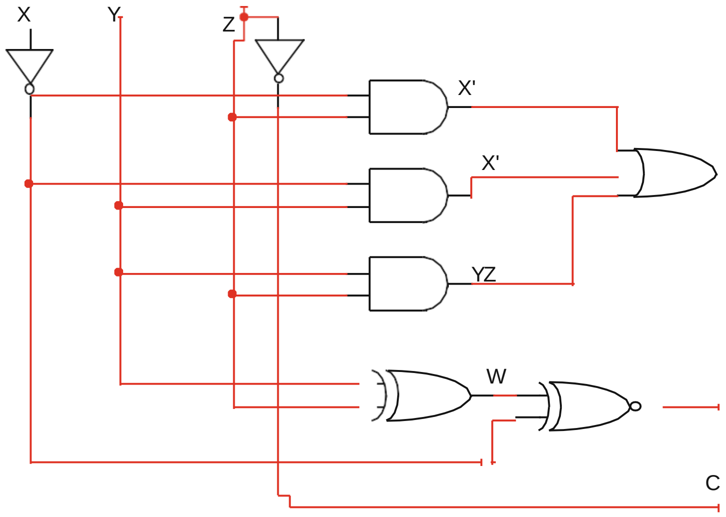
B (X,Y,Z) = $X'y'z' + X'yz + Xy'z + Xyz' = X'(Y'z' + Yz) + X(Y'z + Yz')$

If $Y'z + Yz' = W$ Then

B (X,Y,Z) = $X'w' + Xw$ = X Xnor W And W = Y Xor Z

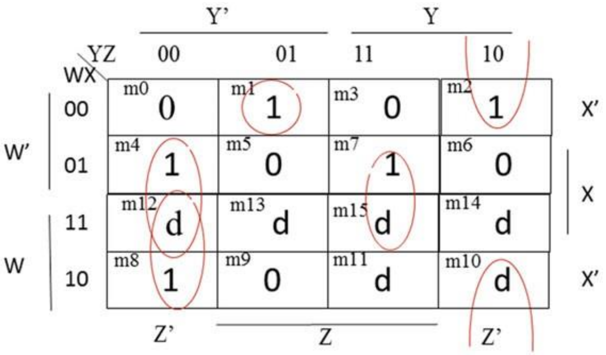
Function For C – By Looking At The Truth Table, The Column For C Is Complement Of Z, Then

C = Z'



۷. د څلورو ورکړیز او یو راکړیز سره یو ترکیبي سرکټ ډیزاین کړئ، ورکړیز د ترکیبي سـرکټ سـره BCD دي، او ورکړیز حتـا پیریټي (Parity) هم د ورکړیز لـه پاره جوړوي.

W	X	Y	Z	F
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	1
1	0	0	1	0
1	0	1	0	d
1	0	1	1	d
1	1	0	0	d
1	1	0	1	d
1	1	1	0	d
1	1	1	1	d



$$F(W,X,Y,Z) = W'x'y'z + Xy'z' + Wy'z' + Xyz + X'yz'$$

۸. څلور بيټه حســابي منطقي برخه (ALU) د لاندي تابع گانو د ترســره كولو له پاره ډيزاين كړئ:

A +B, A - B, A+1, A', B', A Or B, A Xor B, A And B

حل

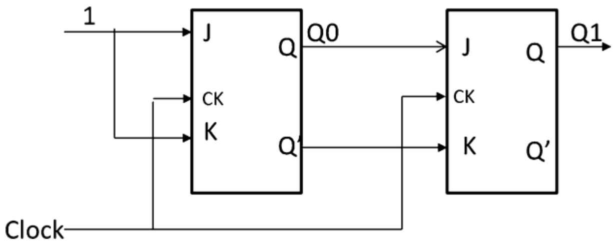
دا يو څلور بيټه حسابي منطقي برخه ده: څکه نو دا څلورو ملټيپلکسرو ته ضرورت لري. حسابي او منطقي برخه اته تابع ګانی لري؛ نو په نتيجه کې هر ملټيپلکسر 8*1 دی.

پنځم څپرکی: مشکلات

۱. لاندي جدول د JK Flip-Flop له پاره مکمل کړئ:

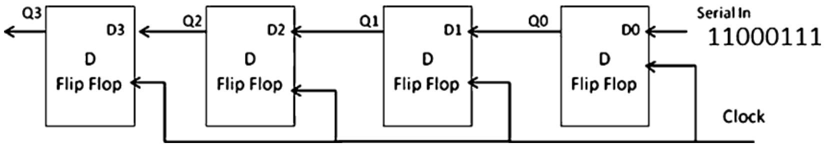
J	K	Q(t) present output	Q(t+1) next output
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

۲. لاندي ارقام يو مسلسل منطق روښانه کوي، لاندي جدول تکميل کړئ، فرض کړئ چې ابتدايي قيمت د Q1=0 او Q2=0. د خپل ځواب د يقيني کېدو له پاره Logicism کاريال استعمال کړئ.



Clock	Q0	Q1
Initial value	0	0
Clock #1	1	0
Clock #2	0	1
Clock #3	1	1

۳. لاندي شـفټ راجسـټر درکړل شـوی، پس له پنځو دوران دساعت (Clock Pulses) راکړيز پيدا کړئ.

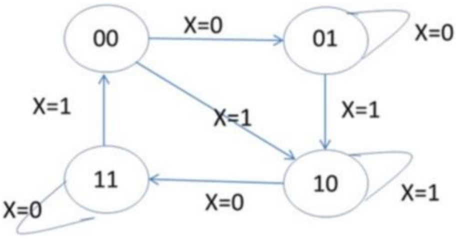


Q3	Q2	Q1	Q0	Clock
0	0	0	0	
0	0	0	1	#1
0	0	1	1	#2
0	1	1	0	#3
1	1	0	0	#4
1	0	0	0	#5

۴. لاندې ايکسټيشن (Excitation) جدول د JK Flip-flop له پاره تکميل کړئ:

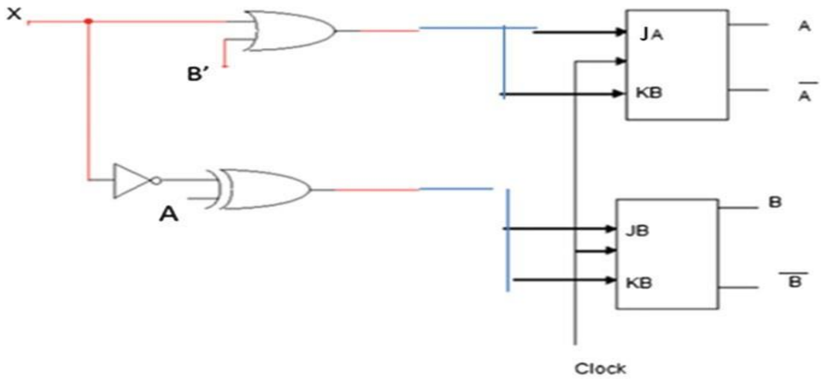
Q(t)	Q(t+1)	J	K
0	0	0	d
0	1	1	d
1	0	d	1
1	1	d	0

۵. د ستيټ ډياګرام (State Diagram) له پاره د ستيټ جدول تيار کړئ:



Present state		Next state		
		X=0		X=1
A	B	A	B	AB
0	0	0	1	10
0	1	0	1	10
1	0	1	1	10
1	1	1	1	00

۶. د لاندي سـرکټ له پاره سـټيټ جدول او سـټيټ ډياګرام (State Diagram) روښـانه کړئ:



Present state		Next state			
		X=0		X=1	
A	B	A	B	A	B
0	0	1	1	1	0
0	1	0	1	1	1
1	0	0	0	0	1
1	1	1	1	0	0

شپږم څپرکی

د بیا کتنې پوښتنې

ګڼ انتخابي پوښتنې

۱. د ــــــــــــ دنده دا ده چې ریاضیکې عملیې اجرا کړي

(a) Bus
(b) Serial Port
(c) ALU
(d) Control Unit

ځواب: C

۲. کله چې تاسو د پروسیس مرکزي واحد (CPU) او کوچنی پروسیسر کارونه سره مقایسه کوي، ــــــــــــ.

(a) سره یو شان دي
(b) سره یو شان نه دي
(c) پروسیس مرکزي واحد (CPU) د کوچنی پروسیسر نه چټک ده
(d) کوچنی پروسیسر د پروسیس مرکزي واحد (CPU) نه چټک ده

ځواب: A

۳. د CISC پروسیسر کنترولونکې برخه ــــــــــــ ده

(a) سخت غالی (Hardware)
(b) مایکروکوډ (Microcode)
(c) a او b
(d) پورته ټول غلط دي

ځواب: A

۴. کوم یو د لاندې 32 بیټه وصلونکې (Buses) دی

(a) Isa

(b) Pci او Eisa
(c) Eisa او Isa
(d) Mca او Isa
ځواب: A

لنډ ځوابه پوښتنې

۱. د پرسیس مرکزي واحد (CPU) دندې تشریح کړئ.

ځواب: پرسیس مرکزي واحد (CPU) د کمپیوټر ماغزه بلل کیږي او د ورکړېزو الاتو له خوا د معلوماتو د تر لاسه کولو مسولیت لري، راکړېز معلوماتو پروسیس کوي، او معلومات د کمپیوټر حافظې او راکړېزو الاتو ته لیږي.

۲. د کنټرول برخه (Control Unit) کومې دندي لري؟

ځواب: د کنټرول برخې وظایف د راکړېزو او ورکړېزو الاتو کنټرولول دي، د کمپیوټر نورو برخو ته کنټرول سیګنالونه تولیدوي لکه د لیکلو او لوستلو سیګنالونه، او د پروګرامونو د اجرا کېدو له پاره لارښوونې ترسره کوي.

۳. په نیم لغت کې څو بیټونه موجود وي؟

ځواب: 16

۴. د کوچنی پروسیسر او د پروسیس مرکزي واحد (CPU) توپیر وکړئ.

ځواب: که چېري د کنټرول برخې راجسترونه حسابي او منطقي برخه (ALU) ټول سره په یوه مدغم شوي سرکټ کې سره بسته شي، دې ته کوچنی پروسیسر وایي: که چېري په یوه سرکټ کې سره بسته نه وي دې ته د پروسیس مرکزي واحد (CPU) وایي.

۵. حافظې ته مستقیم لاسرسي (DMA) دندې تشریح کړئ.

ځواب: DMA (حافظې ته مستقیم لاس رسی) د حافظې نه د ورکړېزو او راکړېزو الاتو ته د معلوماتو د بلاکونو انتقال او د دې برعکس ته اجازه ورکوي. دا پروسه د پروسیس مرکزي واحد (CPU) د کارونی نه غیر ترسره کیږي.

۶. د Serial Port کار څه شی دی؟

ځواب: یو اس بي (USB) چې ډېر زیات کارونه ترسره کوي د Serial Port یو له مهمو قسمونو څخه ده.

۷. د معلوماتو 8 لینونه او ادرسـونو 16 لینونو د پروسـیس مرکزي واحد (CPU) له پاره اعظمی حافظه کومه یوه ده؟

ځواب: پروسیس مرکزي واحد (CPU) د ادرسونو 16 لینونو او معلوماتو 8 لینونو له پاره اعظمی حافظه (8 * 2^16) 524,288 Bits. ده.

۸. د RISC پروسیسر خصوصیات لست کړئ.

ځواب: د RISC پروسـیسـر خصـوصـیات په لاندې ډول دي: نویو لارښـوونو ته ضرورت لري، ټولې لارښووني یو شان اوږد والي لري، اکثره لارښووني په یوه ساعتي دورانی ماشـین کې اجرا کیږي، کنټرول برخه یي سختغالی دی، او دا د یو څو ادرس ورکولو طریقې لري او لوي تعداد کې راجسټرونه لري.

۹. د ملټی کور (Multicore) پروسـیسر فایدې د یو ګوني پروسیسر په مقابل کې څه دي؟

ځواب: ملټی کور (Multicore) پروسـیسر کولی شي چې څو ګوني لارښووني په یوه وخـت کې اجرا کړي چې د دی کـارنـامـه د یو ګوني پروســیســر په مقابل کې زیاتوي.

۱۰. د پیچ (Fetch) لارښود تشریح کړئ.

ځواب: پیچ (Fetch) لارښـود لارښـووني د حافظې څخه د پروسیس مرکزي واحد (CPU) ته انتقالوي.

۱۱. د 21 مشـکل له پاره د Non-Pipeline پروسـیسـر په اسـتعمال سـره د اجرا کولو وخت محاسبه کړئ.

ځواب: 100 ms

۱۲. د ډسک کنټرولونکې اقسام لست کړئ.

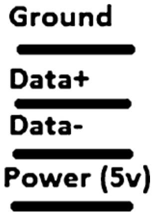
ځواب: دلته د ډسک کنټرولونکې دوه دي، بریښنایي مدغم شوي ډسک (IDE) او د وړوکي کمپیوټر سیستم مخینه (SCSI) .

۱۳. دوه مسلسل وصلونکي (Serial Buses) لست کړئ.

ځواب: یو اس بي (USB) او فیر ویر (FireWire).

۱۴. د یو اس بي (USB) فین وصلونکې وښایي.

ځواب:



۱۵. د فیر ویر (FireWire) کارونه څه شی ده؟

ځواب: فیر ویر (FireWire) د اوچت رفتار مسلسل وصلونکی (Serial Bus) برخه ده کوم چې د ډیجیټل الاتو له پاره استعمالیږي لکه ډیجیټل ویډیو او یا ډیجیټل ویډیو ثبتونکې کمره.

اووم څپرکی: پوښتنی او مشکلات

۱. سم لاسي لاسرسي زېرمه (RAM) د څه شي مخفف ده؟

ځواب: سم لاسي لاسرسي زېرمه (RAM) حافظې ته اټکلي لاس رسي (Random-Access Memory) مخفف ده.

۲. کوم یو د لاندېنو د حافظې د قسمونو څخه د اصلي حافظې له پاره استعمالیږي؟

(a) ROM And SDRAM
(b) SRAM And DRAM
(c) SDRAM And DRAM
(d) DRAM And EPROM

ځواب: C

۳. يواځې لوستونکې زېرمه (ROM) د څه شي مخفف ده؟

ځواب: يواځې لوستونکې حافظه (Read-Only Memory)

۴. د EEPROM او EPROM تر منځ فرق واضح کړئ ؟

ځواب: EPROM دا د بنفشي رڼا (کومه چې له لمر څخه اخستل کيږي) په وسيله له پاکيږي، په داسې حال کې چې EEPROM هغه وخت پاکيږي کله چې يو مخصوص ولتيج په د دې له يوې ستنې سره وصل شي.

۵. د SRAM ابتدايي کا رونه څه دي ؟

ځواب: د SRAM ابتدايي کا رونه د پروسيس مرکزي واحد (CPU) له پاره د کېشي (Cache) حافظې په حيث استعمالېدل دي.

۶. لاندې اصطلاحات تعريف کړئ:

(a) Track
(b) Sector
(c) Cluster

ځواب:

A. ټرکس (Tracks) په دايروي لارو د تريک (Track) تقسيمول دي.

B. سکټر (Sectors) بيا ځلي دتريکس (Tracks) تقسيمول دی په وړو وړو حصو باندې.

C. کلسټر (Cluster) په گروپونو د سکټر تقسيمول دي.

۷. د فايلونو تقسيمونکې جدول (FAT) دنده څه ده؟

ځواب: فايلونو تقسيمونکې جدول(FAT) په هارډسک کې ثبت شوي معلومات تنظيموي، FAT16 او FAT32 به پخوانيو وينډوز کې استعمالېدل.

۸. د کېشي (Cache) اقسام کوم کوم دي ؟

ځواب: د معلوماتو کېشي (D-Cache) او لارښود کېشي (I-Cache).

۹. خيالي حافظه څه شی ده ؟

ځواب: خيالي حافظه اصل کې هارډسک ده او يا کلک حالت لرونکې تخته ده (SSD) چې د عمومي حافظې څخه د کارونې معلومات او لارښوونو د زمېره کولو

لـه پاره اســتعمالېږي کوم چې د پروســیس مرکزي واحد (CPU) ته د اوس له پاره پکار نه وي.

۱۰. فزیکي ادرسونه یو د لاندېنیو اندازه ټاکي

(a) خیالي حافظه

(b) فزیکي حافظه

(c) کېشي حافظه

ځواب: B

۱۱. د تناسب نسبت (Hit Ratio) څه ته وایي؟

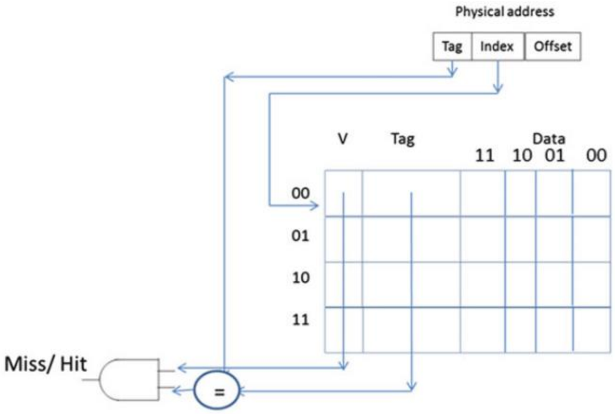
ځواب: د تناسب نسبت (Hit Ratio) دا عبارت ده له، تناسبونو تعدا تقسیم په خطا شوي تعدادونو د تناسبونو باندې.

۱۲. مکانی موقیعت تشریح کړئ.

ځواب: هغـه فکر دی چې وایي کله حافظې یو موقعیت ته لاس رســی وشــي، ډیر احتمال لري چې حافظه کې نژدې ځایونو ته هم اړتیا پیدا شي.

۱۳. د مســتقیمی کړنلاری له پاره د ادرســونو بڼه د کېشــي (Cache) د لید لوري څخه روښانه کړئ.

ځواب:



په خیالي ادرس کې د صفحی لمبر دنده څه شی ده ؟

ځواب: د صفحی لمبر په خیالي ادرس کې د صفحی د پیژندلو له پاره د یوې حصې په توگه استعمالیږي.

16. د صفحو جدول (Table) دنده څه شی ده؟

ځواب: د صفحو جدول د دی له پاره استعمالیږي چې د هر یوي صفحې او د هغی د مربوطه معلوماتو اړوند بلاک څرک وساتي. او همدارنگه صفحی جدول په اصلي او خیالي دواړو حافظو کې موجود د هری صفحې څرک ساتي.

17. د کېشی (Caches) د پیوسته کړنلارې (Associative Mapping) گټې د هغې د مستقیم کړنلارې پر وړاندې څه شی دی؟

ځواب: پیوسته کړنلاره نسبت مستیقیمې کړنلارې ته لږې خطاگانې لري.

18. .... یو تیز ترینه ډول د حافظه ده.

ځواب:

(a) کېشی حافظه

(b) اصلي حافظه

(c) دایمي حافظه

(d) هارډیسک

ستونزي

١. لاندي عمومي او کېشی (Cache) حافظه در کړل شوې. د پروسیس مرکزي واحد (CPU) دا ادرسونه جوړوي.

.0x1, 0x2, 0x1, 0x8, 0x9, 0x1c, 0x1d, 0x3, And 0x4

(a) د دوه طرفه پیوسته کړنلاری په استعمال سره د کېشی (Cache) محتویات روښانه کړئ، د ال ار یو (LRU) بدلولو پالیسي نظر کې ونیسئ.

(b) هیټ ریټ (Hit Rate) څه شی دی؟

Address	Content	Address	Content
00000	5	10000	5
00001	3	10001	0
00010	11	10010	1
00011	6	10011	11
00100	7	10100	15
00101	8	10101	09
00110	9	10110	12
00111	12	10111	23
01000	0	11000	65
01001	0	11001	21
01010	8	11010	8
01011	7	11011	7
01100	9	11100	9
01101	0	11101	0
O1110	2	11110	2
01111	5	11111	5

د ال ار یو (LRU) او وی بیټ (V-Bit) له پاره ابتدایي قیمتونه

Set address	V	Tag	B1	B0	LRU	v	Tag	B1	B0	LRU
00	0				0	0				0
01	0				0	0				0
10	0				0	0				0
11	0				0	0				0

د ادرس پارمټ د کېشي د لیدلوري څخه

2 bits	2 bits	1 bit
Tag	Set address	Byte offset

Set address	V	Tag	B1	B0	LRU	V	Tag	B1	B0	LRU
00	0−>l	00	0	5	0−>−l>−0	0−>1	01	0	5	0−>1
01	0−>1	00	11	3	0−>1	0−>1				0
10	0−>1	11	0	9	0−>−1>−0	0−>1	00	7	6	0−>1
11	0				0	0				0

۲. د 512 بایټه اندازی لرونکي پیج ســره کمپیوټر 32 کېلو بایټه خیالي حافظه او 8 کېلو بایټه عمومی حافظه لري.

(a) څومره بیټونه (Bits) په خیالي ادرس کې موجود دي؟

(b) څومره صفحې په خیالي حافظه کې موجودې دي ؟

(c) څومره بیټونه (Bits) د فزیکي ادرس له پاره پکار دي؟

(d) څومره فریمونه او بلاکونه په عمومي حافظه کې موجود دی؟

(a) 15 خیالي ادرسونه

(b) پیجونه $64= 2^6 = 2^{15} / 2^9$

(c) $8k = 2^{13}$ فزیکي ادرس 13 Bits

۳. کمپیوټر 20 بیټه (Bits) خیالي حافظه لري او د هر پیج اندازه 2kb ده.

(a) د خیالي حافظې اندازه یي څومره ده؟

(b) په خیالي حافظه کې څومره صفحې موجودې دي ؟

ځواب:

(a) 2^{20}=1MB

220/211=29 پیجونه

۴. ۲۴،۷ شکل د پروسیس مرکزي واحد (CPU) 0x00 او 0x0b ادرس جوړوي، فرض کړئ 0 صفحه په یو 1 بلاک کې او 2 صفحه په 0 بلاک کې مپ(Map) شي نو د صفحې جدول محتویات روښانه کړئ؟

ادرس 0x00

Page#	Offset
000	00

صفحې نمبر د صفحې جدول له پاره ادرس دی

0x0b

	Page table	
	Valid bit	Frame number (2 bits)
000	0 1	01
001	0	
010	0 1	00
011	0	
100	0	
101	0	
110	0	
111	0	
	0	

Page #	offset
010	11

010 ادرس دی د صفحې جدول له پاره

اتم څپرکی

۱. د Operands په اساس د هدایاتو ډولونه:

لار ښوونې بیدون له Operand څخه لکه ایچ ال ټي(HLT)

لار ښوونې له دوو Operands سره لکه MOV R1, R2

لار ښوونې له Operands لکه ADD R1, R2, R3

۲. د ستېک فونټر(Stack Pointer) له پاره د ای ار ام (ARM) پروسیسر کوم راجسټر استعمالیږي؟

ستونزې

۳. د R1 محتویات څه شی دي ؟ فرض کړئ R2= 0x00001234.

a) Mov R1, R2, Lsl #4 R1 = 0x00012340
b) MOV R1, R2, LSR #4 R1 = 0x00000123

۴. لاندې HLL کمپیوټري ژبه د ای ار ام (ARM) لارښوونو ته تبدیل کړئ.

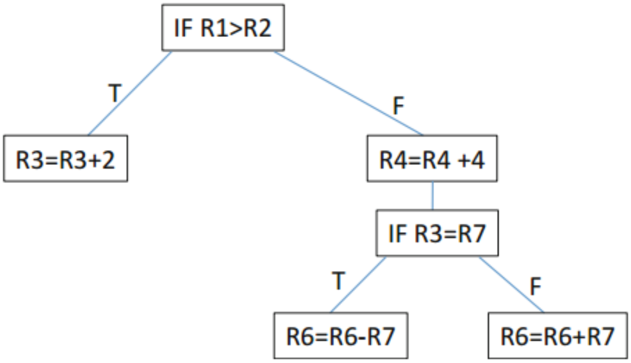
```
If R1>R2 And R3>R4 Then R1= R1 +1
Else
R3=R3 +R3*8

Endif
```

حل:

```
CMP     R1,     R2
CMPGT   R3,     R4
ADDGT   R1,     R1,     #&      1
ADDLE   R3,     R3,     R5      LSL     #3
```

۵. لاندې فلوچارټ د ای ار ام (ARM) اسمبلي ژبي ته تبدیل کړئ.




```
Cmp    R1,    R2
Addgt  F3,    R3,    #&2
Addle  R4,    R4,    #&5
Cmple  R3,    R7
Subeq  R6,    R6,    R7
Addne  R6,    R6,    R7
```

۶. داسي پروګرام ولیکئ چې لاندې ایچ ال ال (HLL) د ARM اسمبلي ژبي ته تبدیل

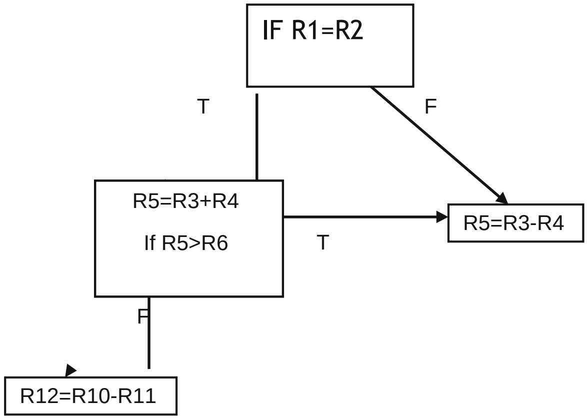
کړئ:

```
a= 10;
b=45;
while (a! =b) {
 if (a < b)
 a=a+5; else b= b+5; }
```

حل:

```
Mov          R1,    #10
Mov    R2,   45
Loop:
 Cmp   R1,    R2
Beq          Halt
Addlt  R1,   R1,    #5
Addgt  R2,   R2,     #5
B Loop
Halt
```

۷. لاندې فلو چارټ د ای ار ایم ARM اسمبلي ژبې ته تبدیل کړئ:




```
CMP    R1,    R2
SUBNE         R5,    R3,    R4
BNE Halt
```

```
ADDEQ     R5,   R3    R4
CMPEQ     R5,   R6
SUBLTE    R12,  R10,  R11
SUBGT     R5,   R3,   R4
Halt
```

نهم څپرکی

ستونزي

۱. لاندې لاربښـوونې تعقیب کړئ، فرض کړئ چې لیسـت د 0x0000018 حافظې له ځایه شروع کیږي او ای ار ام (ARM) لوی انډین (Big Endian) وکاروئ.

ADR R0, LIST ; Load R0 With Address Of Memory Location List

MOV R10, #0x2

(a) Ldr R1, [R0]

(b) Ldr R2, [R0, #4]!

(c) Ldrb R3, [R0], #1

(d) Ldrb R4, [R0 , R10]!

(e) Ldrsb R5, [R0], #1

(f) Ldrsh R6, [R0]

LIST DCB 0x34, 0xf5, 0x32, 0xe5, 0x01, 0x02,0x8,0xfe

حل:

ADR R0, LIST ; Load R0 With Address Of Memory Location List

MOV R10, #0x2

a. LDR R1, [R0]	;R0=0x18	R1=0x34F532E5
b. LDR R2, [R0, #4]!	;R0= 0x1C	R2 =0x010208FE
c. LDRB R3, [R0] , #1	;R0= 0x19	R3=0x34
d. LDRB R4, [R0 , R10]!	;R0= 0x1A	R4=0X32
e. LDRSB R5, [R0], #1	;R0= 0x19	R5=0x34
f. LDRSH R6, [R0]	;R0= 0x18	R6=0x34F5

LIST DCB 0x34, 0xf5, 0x32, 0xe5, 0x01, 0x02,0x8,0xfe

۲. کاري مشـکلات #1 اوله حصـه A او B د وړوکي انډین (Little Endian) په کاروني سره.

(a) R1 = 0xe532f534
(b) R2 = 0xfe080201

۳. د لاندې لوډ لاربښوونو له پاره د Ri راجستر محتویات څه دي، فرض کړئ R0 د وړوکي انډین (Little Endian) په کاروني سره د لست ادرس هم له ځان سره لري.

(a) Ldr R1, [R0]
(b) Ldrh R2, [R0]
(c) Ldrb R3, [R0] , #1
(d) Ldrb R4, [R0]
(e) Ldrsb R5, [R0], #1
(f) Ldrsh R6, [R0]

List DCB 0x34, 0xf5, 0x32, 0xe5, 0x01, 0x02

حل:

(a) LDR R1, [R0] ;R1=0xe532f534
(b) LDRH R2, [R0] ; R2=0x0000f534
(c) LDRB R3, [R0], #1 ;R3=0x00000034
(d) LDRB R4, [R0];R4= 0x000000f5
(e) LDRSB R5, [R0], #1 ; R5=0xfffffff5
(f) LDRSH R6, [R0];R6=0xffffe532

List DCB 0x34, 0xf5, 0x32, 0xe5, 0x01, 0x02

4. د لانـدې لاربښــوونو له اجرا کولو وروســته به موثر ادرس کوم یو وي فرض کړئ R5

شامل د 18 X 0 او R6 شامل 0x00000020 ده.

(A)   Str   R4, [R5]
(B)   Str   R4, [R5, #4]
(C)   Str   R4, [R5, #8]
(D)   Str   R4, [R5, R6]
(E)   Str   R4, [R5], #4

حل:

(A) STR R4, [R5]   EA= 0x18
(B) STR R4, [R5, #4]   EA= 0x18 + 4= 0x1c
(C) STR R4, [R5, #8]   EA=0x18 +8=0x20
(D) STR R4, [R5, R6]   EA= 0x18 +0x20 = 0x38
(E) STR R4, [R5], #4   EA= 0x18, R5=0x18 +4=0x1C

لسم خپرکی

حل:

۱. د ټولو پروګرامونو له پاره، لاندي خاکې استعمال شوي.

```
Area Reset, Data, Readonly
Export __Vectors
__Vectors
Dcd 0x20001000    ; Stack Pointer Value When Stack Is Empty Dcd Reset_Handler ; Reset Vector
Align

 Area Mycode, Code, Readonly

    Entry
```

```
Export Reset_Handler
Reset_Handler
```

ستاسو د کوډ لیکلو ځای

```
Stop   B   Stop
End;            End of the Program
```

۲. یو داسې پروګرام ولیکئ چې لوي عدد پیدا کړي او هغه د حافظې په LIST3 موقعیت کې زیرمه کړئ. فرض کړئ چې اعداد د حافظې په LIST1 او LIST2 په موقعیت کې موجود دي.

```
__main
        ADR   R0,    LIST1
        LDR   R1,    [R0]
        ADR   R0,    LIST2
        LDR   R2,    [R0]
        CMP R1,      R2
        BHI              RESULT  ;  IF R1>R2
        MOV R1, R2
RESULT
        ADR   R0,    LIST3
        STR   R1,     [R0]
LIST1      DCD    0x23456754
LIST2      DCD    0X34555555
LIST3      DCD    0x0
```

۳. یو پروګرام ولیکئ چې دوه اعداد جمع کړي او عدد د لاندې میتود په طریقه وښایي.

```
N1     EQU        5
M1     EQU       7
```

حل:

```
MOV R6, #M1        ; Load    R6        with   7
MOV R8, #N1

_Main
    Add R7, R6, R8         ; Add R6 with R8 and Store
    in R7
 N1          Equ       5
 M1          Equ       7
```

۴. یو پروګرام ولیکئ چې List1 او LIST2 سره جمع کړي، او نتیجه یې په LIST3 کې زېرمه کړي.

```
__Main
    ADR  R1, LIST1 ; ADR Is Pseudo Instruction
    LDR  R2,  [R1]
    ADR  R3,  LIST2
    LDR  R4,  [R3]
    ADD  R5,  R2,     R4
    ADR  R8,  LIST3
    STR   R5 ,  [R8]
LIST1  DCD 0x00002345
LIST2  DCD 0X00011111
LIST3  DCD 0x00000000
```

۵. یو پروګرام ولیکئ چې دوه عددونه سره جمع کړي.

```
__Main
    LDR        R1,      =0x22222222
    LDR        R2,      =0x3
    MUL         R3,     R1,      R2
```

۶. یو داسې پروګرام ولیکئ چې پوسټ انډکس ادرس ورکول (Post-Index Addressing) په استعمال سره اته اعداد سره جمع کړي.

```
LIST DCB 0x5, 0x2,0x6,0x7 ,0x9,0x1,0x2,0x08
```

حل:

```
        ADR   R0,     LIST
__main

       SUB    R5,     R5,    R5
       MOV    R1,     #0x8
LOOP
       LDRB   R2,     [R0],  #1
       ADD    R5,     R5,    R2
        SUB   R1,     R1,    #01
        CMP   R1,     #0x0
        BNE   LOOP
        ALIGN

LIST DCB 0x5, 0x2,0x6,0x7 ,0x9,0x1,0x2,0x08
```

۷. يو پروګرام وليکئ چې ايچ ال ال (HLL) ژبې د ای ار ايم (ARM) لارښـــونو تـه تبديل کړي:

```
IF   R1=R2     AND    R3>R4     then
R1= R1 +1
Else
R3=R3 +R3*8
Endif

ADR R6,  LIST1
ADR R7,  LIST2
ADR R8,  LIST3
ADR R9,  LIST4
   LDR R1, [R6]
   LDR R2, [R7]
```

```
   LDR R3, [R6]
   LDR R4, [R7]

CMP  R1,  R2
CMPEQ  R3,  R4
ADDGT  R1, R1,  # 1
ADDLE  R3,  R3,  R3,  LSL  #3

ALIGN

LIST1   DCD   0x58
LIST2   DCD   0x95
LIST3   DCD    0x24
LIST4   DCD    0x23
STOP   B   STOP
```

۸. . یو داســـې پروګرام ولیکئ چې لاندې ایچ ال ال (HLL) د اســـمبلي ژبې ته تبدیل کړي:

```
If        R1=R2  Then
R3= R3+1
If        R1<R2  Then
R3=R3-1
If         R1>R2          Then
R3=R3-5
```

حل:

```
            MOV R1,       #0x9
            MOV R2,        #0x6
            MOV R3,        #0x5
__main
            CMP R١, R2
            ADDEQ R٣, R3, #0x1
            SUBLE R3, R٣,   #0x1
```

```
ADDGT R۳, R3, #0x3
```

۹. يو پروګرام وليکې چې R1 16 ځله تکرار شي، فرض کړئ چې R1 کې 0x12345678 موجود دي.

```
__main
        ADR R0, LIST
        LDR R1, [R0]
        ROR R1, R1, #16
LIST    DCD 0x1234567
```

۱۰. يو پروګرام وليکې چې د حافظې په LIST موقعيت کې لغت و لولي او د B4 څخه تر B7 پورې د R5 راجستر د بيټ (Bit) موقعيت پاک کړي ، فرض کړئ R5 کې 0xfffffff موجود دي.

```
LDR R0, =0x000000f0
LDR R5, =0xfffffff
```

حل:

```
        LDR R0, =0x000000f0
        LDR R5, =0xffffffff
 __main
        BIC R4, R5, R
```

۱۱. داسې پروګرام وليکئ چې راجسترد R1، R2، R2 او R4 د حافظې له LIST موقيعت څخه لوډ کړئ.

```
LIST DCD 0x12345aaa, 0x0000bbbb, 0x0000cccc , 0X0000DDD

__main
        ADR R0,LIST
        LDM R0, {R1,R2,R3,R4}
```

LIST DCD 0x12345AAA, 0x0000BBBB, 0x0000CCCC ,
0X0000DDD

ماخذونه

1. M. Mano, *Digital design*, 5th edn. (Pearson, 2013) USA
2. E.O. Hwang, *Digital design and microprocessor design with interfacing*, 2nd edn. (Cengage Learning, 2018) USA
3. D. Haris, S. Haris, *Digital Design and Computer Architecture ARM Edition* (Morgan Kaufmann, 2016) USA
4. M. Wolf, *Computers as Components* (Morgan Kaufmann, 2017) USA
5. A. Elahi, T. Arjeski, *ARM Assembly Language with Hardware Experiments* (Springer, 2015) USA
6. W. Stalling, *Computer Organization and Design*, 10th edn. (Pearson, 2016) USA
7. A. Clements, *Computer Organization and Architecture Themes and Variations* (Cengage Learning, 2014) USA
8. NXP Corp, LPC16XX user manual
9. http://infocenter.arm.com, ARM V7 manual
10. Keil Corp, µvision development tool
11. ARM Cortex-M3 technical reference manual
12. S.B. Furber, *ARM System-on-chip Architecture* (Addison Wesley, 2000) USA
13. W. Holm, *ARM Assembly Language* (CRC Press, 2009) USA
14. K. Schindler, *Introduction to Microprocessor Based System Using the ARM processor* (Person, 2013) USA
15. J.W. Valvano, *Embedded Systems Real-time Interfacing to the ARM Cortex-M3* (J.W. Valvano, 2011) USA
16. D. Lewis, *Fundamentals of Embedded Software with ARMCotex-M3* (Pearson, 2013) USA

17. R. Gibson, *ARM Assembly Language – An Introduction*. (LuLu, 2007) USA

انډکس

ال ډی ار (LDR) خیالي لاربښوونې ۱۸۵،۱۸۶
۱۶ بیټه اټکلي د لاس رسی وړ حافظه (RAM) ۱۳۸
32 بیټه (Bit) حسابي کنترولونکې برخه (ALU) ۱۶۵
۳۲ بیټه پروسیسر ۱۱۸
۶۴ بیټه مرکزي عملیاتي برخه ۱۱۸
CISC یو ډول کمپیوټر چې په ټیټه سطحه عملیي په یوه لاربښوونه اجرا کولی شي ۱۱۸، ۱۱۹
۵۲ Function F(X, Y),
۱۹۸ μVision
اتومات ډول په راجسټر کې د ادرس زېرمه کولو طریقه ۱۸۸،۱۸۹
اتومات ډول په راجسټر کې د ادرس زېرمه کېدل ۱۹۰
اجزاوې ۱۱۵،۱۱۶
ادرس ورکولو طریقه ۱۸۷
ادرس ورکولو طریقه، ای ار ام (ARM) ۱۸۷
ار (OR) عملیه ۳۰
ار (OR) ګېټ ۳۱

ار ډي ار ای ام سملاسي لاسرسي زېرمه (RDRAM) (یو ډول حافظه) ۱۴۰
ار (OR) منطق ۳۰
اس ار لیچ ۹۸،۹۹
اس ټي ار (STR) لاربښوونې ۱۸۷
اس ډي ار ای سملاسي لاسرسي زېرمه (SDRAM) (یو ډول حافظه) ۱۴۰
اسکي (ASCII) کوډ ۱۷،۲۵
بلې بڼې ته بدلوونکی ۷۵،۷۶

اصلي بڼې ته بدلوونکی ۷۲
اصول ۳۶، ۳۷
اعشاري اعدادو برخه ۱۴،۱۵
اعشاري اعدادو ښودنه ۱۴،۱۵
کاریال ۴۸
اکس او ار (XOR) ګیټ ۳۲
انالوګ سیګنالونه ۴
انسټالونکی بنډل ۱۹۸
اوچته روښانونکې ویډیوي صفحه (HDMI) ۱۳۳،۱۳۴
اوو برخو والا مخ پاڼه ۸۹
اي ډي او ار ای ام (EDORAM) (یو ډول ځغلنده حافظه) ۱۴۰
ای ار ام (ARM) راجسټر ۱۶۵،۱۶۶
ای ار ام ۷ نسل (ARMv7) اسمبلي پروګرام ۱۹۹
انډ (AND) ګیټ ۳۰،۳۱
انډ (AND) لاجیک ۳۰،۳۱
انډ ناټ (AND-NOT) ګیټ ۳۲
بایټ (Byte) ۵
بریښنایي سیګنال ۳
دبریښنا په ذریعه لمنځه تلونکی یوازې لوستونکې حافظه (EEPROM) ۱۴۱
بریښنایي مدغم شوي دړه ۱۲۵
بلاک (Block) ۱۴۶
بلاک ډیاګرم ۱۳۸
بولین منطق ۳۰
بي ثباته حافظه ۱۳۸
بي علامې اعداد ۱۲

بیټ (Bit) ۵
پاکولو وړ د پروګرام کولو وړ یوازې د لوستلو وړ حافظه ۱۴۱
پرله پسې ادرس ورکول ۱۸۸
پرله پسې لېږد ۲۴
پرمختللی مسلسل تکنالوجيکې ضمايم (SATA) ۱۲٦
په اسانه طریقه د ادرس زېرمه کېدل ۱۸۹،۱۹۰
په امریکایي معیار د معلوماتو او کوډ مبادله کول (ASCII) ۱۸
په باره کې ځای ښودونکی (SP) ۱٦۵،۱۷۷،۱۸۱
په حافظه کې د ډیټا ځای مختص کوونکی ۲۱۰،۲۱۱
په حافظه کې د یو یا ډیرو نیمه لغاتونو له پاره ځای مختص کول ۲۱۰
په حافظه کې د یوی کلمي د بیټونو ترتیبول ۱۹۳
په خاص لمبر سره د معلوماتو د لغت بدلوونکی ۱۰۲،۱٦۵
په ډیره لویه پیمانه مدغم کېدل (VLSI) ۲۹،۳۳،۳٦
په راجسټر کې د معلوماتو لیکل ۱٦۵
په کمه کچه مدغم کېدل ۳۴
په کوچنۍ کچه ادغام ۳۴
په لویه کچه ادغام ۳٦
په منځني کچه ادغام کېدل (MSI) ۳٦
پیپ لین ۱۲۴
تاق مساوات ۲۱
تام عدد ۱۴
تحلیل ۷۲، ۷۳
ترکیبي لاجیک ۸۸، ۹۰
تشریح ۱٦۳،۳٦،۷۱
ټرانزسټر ۳۳

ټېسټ کونکی / فرضی لابراتوار، چلول ۲۰۴،۲۰۶
ټي فليف فلاف ۱۰۱
ټی فليف فلاف د معلوماتو جدول ۱۰۸
جدول کې د خانو يو ځای والی ۵۵
جفت مساوات ۲۱
جوړښت ۱۲۰
جی کی فليف فلاف ۱۰۰،۱۰۲
جی کی فليف فلاف د معلوماتو جدول ۱۰۸
ځانګړی ار ګيټ (OR Gate) ۳۲
ځانګړی نار ګيټ (NOR Gate) ۳۳
ځايي موقيعت ۱۴۶
چپ طرف ته منطقي بدليدل (LSL) ۱۷۱
څلور متغير ۵۷
څو هسته لرونکی پروسيسر ۱۲۲، ۱۲۳
حافظه ۲
حافظې ته د لاس رسي وخت ۱۴۱
حافظې ته مستقيم لاس رسی (DMA) ۱۲۴
خيالي ادرس ۱۴۶
خيالي حافظه ۱۵۳،۱۵۴
خيالي لارښوونې ۱۸۵، ۱۸۶
د B او BL لارښوونو خاکه ۱۷۹
د IEEE له خوا جوړ شوو سټنډرډونو مجموعه ۲۱۰
د ادرسونو ليږونکی ۱۱۷
د اضافي وسايلو د اتصال وصلونکی۱۲۷
د اضافي وسايلو په بشپړه توګه وصلول ۱۳۲

د اضافي وسایلو په بشپړه توګه وصلولو د اصولو جوړښت ۱۳۲
د اضافي وسایلو د اتصال بشپړول ۱۳۱،۱۳۲
د اعدادو سیستم ۵
د امبارولو عملیه او لارښوونې ۱۷۷،۱۷۸
د انټل د پروسیسرونو کورنۍ ۱۲۰،۱۲۲
د انټل کمپنی مایکرو پروسیسر ۱۲۰، ۱۲۲
د اوو خطونو په ذریعه د اعدادو ښودنه (لکه ډیجیټلی ساعت) ۸۹،۹۱
د اهتزازي حرکت پراخوالی ۴
د ای ار ام (ARM) اسمبلي پروګرام ژبې د پروګرام کولو کړنلاره ۲۰۸
د ای ار ام (ARM) خیالي (Pseudo) لارښوونې ۱۸۶
د ای ار ام (ARM) ډیټا پروسیس کولو لارښوونې ۱۷۴،۱۷۵
د ای ار ام (ARM) ډیټا لیږلو لارښوونې ۱۸۵
د ای ار ام (ARM) لارښوونې ۱۶۹
د بدلولو راجسټر ۱۰۱
د بدلون قانون ۳۸
د بدلیدلو او څرخولو لارښووني ۱۷۱،۱۷۲
د بلاک ډیاګرام ۷۲
د بولین اصول ۲۹
د بولین الجبر د اصولو د همغږۍ قانون ۳۸
د بولین وظایف ۳۹،۴۰
د پروسیسر حمایه کول ۱۸۷
د پروګرام خاکه ۲۰۸
د پروګرام د حالت معلوموونکی راجسټر ۱۶۵،۱۶۶
د پروګرام کولو او تشولو وړ موقتي حافظه (EEPROM) ۱۴۱
د پروګرام کولو وړ موقتی حافظه (EPROM) ۱۴۱

د پروګرام یو ځای کېدل ۲۰۱
د صفحو جدول ۱۵۴
د تقسیمولو اصول ۳۷
د تولیداتو جمع کول (SOP) ۵۸
د ثابت لغات تشریح ۲۱۰
د جمع حاصل ۵۸،۶۰
د حافظې عملیې ۱۴۵
د حافظې او راجستر تبدیلونکی (SWAP) ۱۹۱
د حافظې تسلسل کېشی حافظه ۱۴۵،۱۴۶
د حافظې موقعیتونه ۱۳۹
د حروفو جمله ۲۱۱
د حسابي لاجیک برخه (ALU) ۱۴،۸۸،۹۰
د درېو ورکړیزو (Input) جمع کونکی ۸۴،۸۶
د دوو په قاعدې (Binary) کلاس سره د لسو د قاعدې د اعدادو بدلول بلی بڼې ته ۱۶،۱۷
د دوو په قاعدې اعدادو منفي کول ۸۷
د دوو څلورو بیټونو د چپ طرف راجستر بدلول ۱۰۳
د دوو څلورو بیټونو والا باینری (Binary) اعدادو جمع کول ۸۷
د دوو قاعدې د اعدادو جمع ۱۰
د دوو له قاعدې څخه د لسو قاعدې (Decimal) ته بدلول ۶-۷
د دوو ورکړیزو (Input) جمع کونکې ۸۴،۸۵
د ډسک فایل سیستم ۱۴۳
د ډسک کنټرولونکی ۱۲۵،۱۲۶
د ډی فلیف فلاف سټیټ ډیاګرام ۱۰۶
د ډیټا پروسیس کولو لارښوونې ۱۶۷،۱۶۸

د ډيټا ښودل او حافظه ۱۹۲،۱۹۳،۲۰۹
د ډيټا لوستلو راجسټر ۱٦۵
د ډيټا وصلونکی ۱۱۷،۱۱۸
د ډيجيټل وسايلو ترمنځ د معلوماتو لېږدونی ميتود ۱۳۲
د ډيمورګن (De Morgan) لومړی قانون ۳۸
د ډيمورګن (De Morgan) دوهم قانون ۳۸
د راجسټر بانک ۱٦۴
د راجسټر بدلولو لارښوونې (MOV and MVN) ۱۷۰
د راحسټر ادرس ۱٦۴
د رمبس کمپنۍ هممهاله متحرک رم (RAM) ۱۴۰
د پوستغالو طبقه بندي ۱۳۲
د پروسيس مرکزي واحد (CPU) وصلونکی ۱۱٦
د شپاړلسو په قاعدې اعداد ۹،۱۷،۲۵
د شپاړلسو قاعدې اعدادو څخه تر دوو قاعدې اعدادو پورې ۹
د نه پاملرنی حالت ٦۰،٦۱
د ضريب غونډول ۱۸۰
د طاقت (قاعدې) روښانونکی ۱۵،۱٦
د عملياتي لارښوونو نه شتون ۱٦۲
د عمومي مسلسل اتصال اله ۱۳۰
د عمومي مسلسل وصلولو جوړښت ۱۲۹
د عمومي مسلسل وصلولو کيبل ۱۳۰
د فابريکې معياري جوړشوی وصلونکي ۱۲۷
د فليف فلاف د معلوماتو جدول ۱۰۷
د کارونکي د لاس رسي وخت حافظې ته کموي ۱۵۵
د کمپيوټر اساسي برخي ۲

د کمپیوټر د پروګرام لیکلو له پاره د معلوماتو د جوړښت داخلول ۱۹۲
د کمپیوټر د پروګرام لیکلو له پاره د معلوماتو د جوړښت داخلولو لارښوونې (BFI) ۱۹۲
د کمپیوټر د پروګرام لیکلو له پاره د معلوماتو د جوړښت لمنځه وړل (BFC) ۱۹۱
د کمپیوټر د پروګرام لیکلو له پاره د معلوماتو د جوړښت لمنځه وړلو لارښوونې (BFC) ۱۹۱
د کوچني کمپیوټر سیستم ۱۲۵
د کوډ کولو طرحه اسکي (ASCII) کوډ ۱۷
د کوډ ګروپ کول ۱۷۳
د منطقي ګیټ عملي کېدل ٦۴
د لارښوونو اجرا کول ۱۲۳
د لارښوونو بل بڼې ته اړونکي او منطقي کنټرول ۱٦۴
د لارښوونو چالانیدل (LDR) ۱۸۵
د لارښوونو خاکې او عملیې ۱٦٦
د لارښوونو د مجموعو د جوړښت تفصیل (ISA) ۱٦۲
د لږې لارښووني سیټ والا کمپیوټر (RISC) ۱۱۹
د لسو قاعدې اعدادو تبدیلول د دوو قاعدې ته ۷،۹
د لوړې سطحې د پروګرام جوړولو ژبه (HDMI) ۱٦۱
د لېږد طریقه ۲۲
د لېږد میتود ۲۴
د مدغم شوي سرکټ د ستنو شمیر ۳۵
د مسلسل لاجیک تحلیل ۱۰۴-۱۰٦
د مسلسلې پرمخ تللي تکنالوجی ضمایم(SATA) ۱۲٦
د معلوماتو د لېږد له پاره د دوه ستنو والا چپف (یود ډیټا د لیکلو یو د ډیټا د لوستلو له پاره) ۱۳۹
د مقایسې او ټسټ کولو لارښوونې ۱٦۹

د منفي عملیي ساده کونکی ۱۱،۱۲
د مایکرو پروسیسر او مایکرو کنټرولر له پاره پروګرام جوړونکی ۱۹۷
د مایکروسافټ انسټالونکی ۳٦
د مینټرمس (Minterms) لېږدول ۵۳
د نار (NOR) ګیټ په استعمال سره ٦۳،٦۵
د نانډ (NAND) ګیټ په استعمال سره ٦۲،٦۴
د ورکړیزو او راکړیزو مزاحمتونو پروګرام کول ۱۲۵
د ورکړیزې نمونې جوړول ۲۰۱
د وړوکي چېنل وصلونکی ۱۲۷
د وړوکې کمپیوټر وصلونکی ۱۲٦
د وصل کېدو لپار یو ډول عمومي سوری ۱۲۷
د وصلولو بنیادي نقطه ۱۳۰
د وصلولو مرکزي نقطه ۱۳۰
د وصلونکو کېبلونو کنټرولونکی ۱۱۸
د هارډ وارډ پوهنتنون له نظره د کمپیوټر جوړښت ۱۲۰
د هارډویر جوړښت ۱۲۰
داخلي جوړښت ۱۴۲
دري خانه یي اله ۳۳
دري متغیره ۵۳،۵۴
دوران ۲۱، ۲۲
دوراني ځنډ ۱۴۳
دوراني کلستر ۱۴۳
دورانی سګنل ۲
دوره یي سیګنل ۲،۳
دوه برابره دقت والې ۱۵

دوه عملياتي لارښوونې ۱۶۳
دوه ييز کمپليمنټ ۱۲
دوهمه حافظه ۱۴۵
دويډيو بريښنايي معيار د ټولنو اتصال ۱۲۷
ډولونه ۱۳۹
ډی پليف پلاف ۹۹،۱۰۱
ډيجيټل سيګنلونه ۱،۵
ډيزاين ۷۲
زېرمه شوي لارښوونې (STR) ۱۸۷
راجسټر ۱۰۱،۱۱۶،۱۶۳
رياضيکې جدول ۷۹
ساده شوی تابع ۵۸
سټک پواينټر (SP)۱۶۵،۱۷۷،۱۸۱
سره له څانګې اړيکتيايي لارښوونې ۱۷۸
ګېونکي ۱۰۸، ۱۰۹
ښکاره کونکی ۱۵
ښي طرف ته حسابي بدلون ۱۷۲
ښي طرف ته منطقي بدليدل (LSR) ۱۷۱
ضريب ,(MUL) ۱۸۰
طريقه ۱۶۵
عمومي حافظه ۱۴۵
عمومي ګيټ ۶۱
عمومي مسلسل وصلونکی (USB) ۱۲۸
غونډونکی (Assembler) ۱۹۷
غير دوراني سګنل ۲

غير مشروط او مشروط لاربنووني ۱۷۳،۱۷۴
فاصله ۲۱۱
فاصله ۲۱۱
فريکونسي (F) ۳
فريکونسي (F) ۳،۴
فزيکې ادرس ۱۴۶
فلش (Flash) حافظ ۱۴۱
فلش حافظه ۱۴۱
تابع (Function) ۵۵
کرنف جدول کې نژدی خانې ۵۲،۵۴
کلک حالت لرونکي اله ۱۴۴
کنترولونکې برخه ۱۱٦
کوچني کمپيوټر سيستم وصلونکی (SCSI) ۱۲۵
کېشی (Cache) حافظه ۱۴۵
کېشی (Cache) حافظه او عمومی حافظه وصلونکی ۱۴٦
کېل (Keil) د جوړولو اسباب ای ارام کورټکس M3 پروسيسر ۱۹۹
کېل ليد لوری ۲۲،۱۹۸
کېل ويب سايټ ۱۹۸
لاجيک گيټ ۲۹
لاربنوونې ۲۰۹
لاربنوونې ۱۸۰
لوگاريتم ۱۵،۱٦
له دوو قاعدې اعدادو تر لسو قاعدې اعداو پورې ٦-۷
له لسو قاعدې تام اعدادو تر دوو قاعدې اعدادو پورې ۷
له لسو قاعدې کسري اعدادو تر دوو قاعدې اعداو پورې ۸-۹

ماشين RISC پر مخ تللی (ARM)١٦۴

مدغم شوی سرکټ (IC) ٣۵

مربوطه نقشه ترتيبول ١۵٠

مرحله ۴

مرکزي عملياتی برخه (CPU) ٢،١۴،١۵

مرکزي کنترولر ١٢٩

مستقيم نقشه کشي ١۴٦-١۴٨

مسلسل لاجيک ٧١

مسلسل لېږد ٢۴

معياري ١١۵

ملټيپلکسر (MUX) ٧٧،٨١،٨٣

منطقي رياضيکي برخه (ALU) ١١٦

منفي کونکی ٨٨

موازي لېږد ٢۴

يوازي لوستونکی يا موقتي حافظه (ROM)

موقتي ځای ١۴٦

ميکسټرم ۵١

مينټرمس (Minterms) ۵٢

ناټ (NOT) ګيټ ٣١

ناټ (NOT) لاجيک ٣١

نار (NOR) ګيټ ٣١

نار (NOR) لاجيک ٣١

نانډ (NAND) ګيټ ٣١

نړيوال کوډ (Unicode) ١٧،٢٠

نورمال شوی لوګاريتم ١۵

نیمه هادي حافظه ۱۳۷
وان نیومان (Von Neumann) له نظره د کمپیوټر جوړښت ۱۲۰
وان نیومان او هارډورډ له نظره د کمپیوټر جوړښت ۱۲۰
وخت لټول ۱۴۳
ورکړیزه اله ۲
ورکړیزه اله ۲
ورکړه یز نانډ (NAND) ګیټ ۳۱
ورکړه یزې او راکړیزې الې ۱۲۵
کوچنی کمپیوټر ۱۱۶
وقفه یي لیږد ۲۲
هارډسک ته د لاس رسي وخت ۱۴۳
هب جوړښت (د یو مخصوص چپ ډیزاین)۱۳۰
هټ (Hit) تناسب۱۴۶
هغه تخنیک دی چې د کېشـــي (Cache) حـافظې له پاره په عمومي حافظه کې خالي ځایونه نقشه کوي ۱۵۲
هغه سرکټ چې د حافظو له ډسک څخه د ډیټا لیږد را لیږد کنترولوي ۱۲۵
هم مهاله اس ډي ار ای ام سملاسي لاسرسي زېرمه (SDRAM) ۱۳۹
هم مهاله لیږد ۲۳
هم مهاله مسلسل لاجیک ۱۰۴
یو حرف ۲۱۱
سملاسي لاسرسي زېرمه (RAM) ۱۴۰
یواځنی حرف ۲۱۱
یواځنی دقت والې ۱۵
یوه عملیاتي لارښوونه ۱۶۳

د بادام نیازي پېژندنه

ښاغلی بادام نیازي د ګل جان زوی د لغمان ولایت د علینګار ولسوالی د سنګر کلي اوسیدنکی دی.

په ۲۰۰۶ کال د ننګرهار پوهنتون د کمپیوټر ساینس پوهنځي څخه په کادر کې فارغ او په همدې کال بېرته په همدې پوهنځې کې د استاد په توګه په دنده و ګمارل شو.

په ۲۰۱۲ کال د بورس له لارې پيښور پوهنتون ته کامیاب او په ۲۰۱۵ کال کې فارغ شو.

ښاغلی نیازي د استادي تر څنګ ٦ کاله د ژغورنې نړیوالې موسسې د ختیځ زون د ښوونې او روزنې د مسوول په توګه هم دنده اجرا کړې ده.

د ننګرهار پوهنتون له خوا د هېواد څخه بهر په مختلفو کنفرانسونو او ورکشابونو کې د پوهنتون د استازي په توګه هم ګدون کړی دی او د نیازي صائب ډېری مقالې په بېلابېلو نړیوالو ژورنالونو او کنفرانسونو کې خپرې شوي.

نشرشوی مقالي

TITLE	CITED BY	YEAR
TetraMail: a usable email client for blind people *A Khan, S Khusro, B Niazi, J Ahmad, I Alam, I Khan* *Universal Access in the Information Society 19, 113-132*	17	2020
Energy-efficient cluster formation in IoT-enabled wireless body area network *A Zeb, S Wakeel, T Rahman, I Khan, MI Uddin, B Niazi* *Computational intelligence and neuroscience 2022*	11	2022
A touch sensitive keypad layout for improved usability of smartphones for the blind and visually impaired persons *B Niazi, S Khusro, A Khan, I Alam* Artificial Intelligence Perspectives in Intelligent Systems: Proceedings of …	10	2016

TITLE	CITED BY	YEAR
Automated detection of rehabilitation exercise by stroke patients using 3-layer CNN-LSTM model *ZU Rahman, SI Ullah, A Salam, T Rahman, I Khan, B Niazi* *Journal of Healthcare Engineering 2022*	8	2022
Evaluating smartphone screen divisions for designing blind-friendly touch-based interfaces *S Khusro, B Niazi, A Khan, I Alam* *2019 international conference on frontiers of information technology (FIT …*	4	2019
Interest-based content clustering for enhancing searching and recommendations on smart TV *M Jan, S Khusro, I Alam, I Khan, B Niazi* *Wireless Communications and Mobile Computing 2022*	2	2022
Towards a Low-Cost Teacher Orchestration Using Ubiquitous Computing Devices for Detecting Student's Engagement *I Ahmad, S Khusro, I Alam, I Khan, B Niazi* *Wireless Communications and Mobile Computing 2022*	1	2022
Smartphone-Based Cognitive Assistance of Blind People in Room Recognition and Awareness *M Ullah, S Khusro, M Khan, I Alam, I Khan, B Niazi* *Mobile Information Systems 2022*	1	2022
Improving Security Architecture of Internet of Medical Things: A Systematic Literature Review *M Mahmood, MI Khan, H Hussain, I Khan, S Rahman, M Shabir, B Niazi* *IEEE Access*		2023
Research Article Automated Detection of Rehabilitation Exercise by Stroke Patients Using 3-Layer CNN-LSTM Model *ZU Rahman, SI Ullah, A Salam, T Rahman, I Khan, B Niazi*		2022

Abstract

I am grateful to Almighty Allah for giving me the chance to translate an essential textbook pertaining to my profession into my native language and thereby serve my nation and fellow citizens.

The translated book is written by Ata Elahi and is titled Computer Systems Digital Design, Fundamentals of Computer Architecture, and Assembly Language. In 2018, it was published and taught at Southern Connecticut State University in New Haven, CT, USA. For the purpose of academic rank promotion, the Database and Information System Department, Faculty of Computer Science, Nangarhar University, assigned me to translate this book into the native (Pashto) language under the direct supervision of Associate Professor Sediqullah Barakzai of the Faculty of Computer Science, Kabul Education University. It was presented at the meeting of the Department of Integration of the Ministry of Higher Education on Jadi 17, 1398, of the Commission for the Selection of Academic Topics for Professors and registered with Ref. No. (643) in the Research, Compilation, and Translation Database.

In addition, I did my best to adhere to the rules of translation in protecting the author's and the book's rights, and I translated it into my native Pashto language without changing any words.

Fortunately, the translated book comprises more than 80 parts. The digital logic design subject is taught as a specialty subject in the database and information systems department. Based on this, it will be highly useful for the department's students and lecturers.

Finally, I would like to express my gratitude to Associate Professor Siddiqullah Barakzai, who served as my supervisor, as well as to all of my friends who helped me translate this book and provided the necessary guidance.

Regards
Assistant Prof.: Badam Niazi

د افغانستان د ۹ پوهنتونونو د ۳۸۹چاپ شویو درسي کتابونو لېست

(کابل، کابل طبي پوهنتون، کابل پولي تخنیک، ننگرهار، خوست، کندهار، هرات، بلخ او کاپیسا) ۲۰۱۰ ـ ۲۰۲۳

شمېره	د کتاب نوم	لیکوال	پوهنتون	شمېره	د کتاب نوم	لیکوال	پوهنتون
۱. اخلاق، طبي لارښود او ترمینولوژي							
۱	اخلاق طبابت	پوهاند داکټرعبدالغفورهمدل صدیقی	بلخ	۲	رهنمای تدریس طب	پوهاند دوکتورنادراحمد اکسیر	کابل طبي پوهنتون
۳	د طبابت لنډ تاریخ	پوهاند عبدالحی مومنی	ننگرهار	۴	طبی ترمینولوژي	دوکتورگل سیما ابراهیم خیل قادری	ننگرهار
۵	د ننگرهار طب پوهنځی نصاب او درسي مفردات (انگلیسی)	ننگرهارطب پوهنځی	ننگرهار	۶	رهنمود PBL درافغانستان	پوهنوال دوکتورمحمد فرید برنایار	کاپیسا
۷	انگلیسي- پښتو طبي قاموس I	رنځورمل دوکتور عجب گل مومند	ننگرهار	۸	انگلیسي- پښتو طبي قاموس I	رنځورمل دوکتور عجب گل مومند	ننگرهار
۹	د طب محصلینو درسي کتابونه	ډاکتر یحیی وردک	ټول پوهنتونونه	۱۰	۱۴۰ طبي کتابونه په ډي وي ډي کې (پښتو، دري او انگلیسي)	بېلا بېل مؤلفین	ټول پوهنتونونه
۱۱	۲۱۴ طبي کتابونه په ډي وي ډي کې (پښتو، دري او انگلیسي)	بېلا بېل مؤلفین	ټول پوهنتونونه	۱۲	د طبیعي علومو انگلیسي- پښتو قاموس	پوهنوال ډاکتر نظر محمد سلطانزی خدران	ننگرهار
۲. فزیک							
۱۳	په معاصر طب کې د فزیک پېژندنه	گل احمد سهیل	ننگرهار	۱۴	بیوفزیک	پشتانه بنایي	ننگرهار
۱۵	بیوفزیک	پوهنیار گل احمد سهیل	ننگرهار	۱۶	بیوفزیک	پوهاند میر محمد ظاهرحیدری	بلخ
۱۷	طبي فزیک	پوهنیار هدایت الله مهمند	ننگرهار	۱۸	فزیک طبی بخش میخانیک	پوهاند میر محمد ظاهرحیدری	بلخ
۱۹	فزیک طبی بخش حرارت	پوهاند میر محمد ظاهرحیدری	بلخ	۲۰	توضیح اساسات فزیکی، وسایل تشخیصیه طبی	پوهاند میر محمد ظاهرحیدری	بلخ
۲۱	فزیک نور	پوهاند میر محمد ظاهر حیدری	بلخ	۲۲	فزیک اوپتیک	پوهنوال غلام قادر دهگان	هرات
۲۳	نور و فزیک جدید	پوهنوال غلام قادر دهگان	هرات	۲۴	د نور فزیک	پوهنیار هدایت الله مهمند	ننگرهار
۲۵	د برق فزیک	پوهنیار هدایت الله	ننگرهار	۲۶	میخانیک او د نور فزیک	پوهنیار هدایت الله	ننگرهار
۲۷	کوانتم میخانیک	پوهنیار اکرام الله وقار	ننگرهار	۲۸	حرارت و ترمودینامیک	پوهنوال غلام قادر دهگان	هرات
۲۹	برښنا، مقناطیسیت او الکترو مقناطیسي تیوري	پوهندوی توریالی همدرد	ننگرهار	۳۰	میخانیک، اهتزازات او نسبیت	پوهندوی علي جان عادل	ننگرهار
۳. کیمیا							
۳۱	طبي کېمیا	پوهنوال امرالله آصفي	خوست	۳۲	طبي بیوشمي	پوهاند خان محمد احمدزی	کابل طبي پوهنتون
۳۳	کېمیایي عنصرونه، لومړی ټوک	محمد طاهر کاښی	ننگرهار	۳۴	کېمیایي عنصرونه، دوهم ټوک	محمد طاهر کاښی	ننگرهار
۳۵	فزیکي کېمیا گازونه او کېمیاوي ترمودینامیک	پوهاند خیر محمد ماموند	ننگرهار	۳۶	فزیکي کیمیا دوهم جلد، ترمودینامیک	حبیب الله نوابزاده	ننگرهار
۳۷	عضوي کېمیا، کړیوال ترکیبونه	پوهاند دوکتور محمد غوث حکیمي	ننگرهار	۳۸	فزیکي کېمیا II،	پوهاند دوکتور خیر محمد ماموند	ننگرهار
۳۹	فزیکي کېمیا III، کېمیاوي کنتک او کتلسس، کروماتوگرافي او اسپکتروسکوپي	پوهاند دوکتور خیر محمد ماموند	ننگرهار	۴۰	عمومي کیمیا	پوهاند دوکتورخیرمحمد ماموند	ننگرهار
۴۱	عضوی کیمیا، د الیفاتیک برخه	پوهنوال ډاکترگل حسن ولیزی	خوست	۴۲	عضوي کیمیا، د اروماتیک او هیترو سیکلیک برخه	پوهنوال ډاکترگل حسن ولیزی	کابل
۴۳	د قندونو هضم، جذب او استقلاب	پوهیالی دوکتوریحیی فهیم	ننگرهار	۴۴	د شحمیاتو استقلاب	دوکتور محمد عظیم عظیمي	ننگرهار
۴. بیولوژي او جنیټیک							
۴۵	عمومي بیولوژي	پوهندوی جماعت خان همت	ننگرهار	۴۶	عمومی بیولوژي	پوهندوی الفت شیرزی	ننگرهار
۴۷	بیولوژي مالیکولي حجره، بخش اول	پوهنوال علی یوسف پور	کابل طبي پوهنتون	۴۸	بیولوژي مالیکولي حجره، بخش دوم	پوهنوال علی یوسف پور	کابل طبي پوهنتون
۴۹	د حجرې بیولوژي	پوهندوی جماعت خان همت	ننگرهار	۵۰	مالیکولي بیولوژي	پوهندوی جماعت خان همت	ننگرهار
۵۱	وراثت	پوهنوال دوکتور گل سالم شرافت	ننگرهار	۵۲	کلاسیک اومالکیولی جنتیک	دوکتورمحمد صابر	ننگرهار
۵۳	وراثت او د سمورفولوژي	پوهنمل ډاکتر مسیح الله مسیح	ننگرهار	۵۴	طبي جنیتیک	پوهندوی الفت شیرزی	ننگرهار
۵۵	زولوجي فقاریه	ذاکره بابکرخیل	ننگرهار	۵۶	زولوجي غیرفقاریه	ذاکره بابکرخیل	ننگرهار
۵۷	حیوانات مفصلیه	پروفیسور داکتر دیپلوم علی آقا نحیف	هرات				

۵. اناټومي او هستالوژي							
۵۸	اناټومي لومړی جلد (هډوکي، مفاصل او عضلات)	پوهنمل ډاکتر محمد ناصر نصرتي	ننگرهار	۵۹	د هډوکو او مفاصلو اناټومي	پوهنوال دوکتور حميدالله حامد	خوست
۶۰	د سر او غاړې اناټومي درسي کتاب I	پوهندوی دوکتور یما صديقي	ننگرهار	۶۱	د سر او غاړې اناټومي درسي کتاب II	پوهندوی دوکتور یما صديقي	ننگرهار
۶۲	د ټټر اناټومي	پوهنیار دوکتور یما صديقي	ننگرهار	۶۳	اناتومی	پوهنمل ډاکټرحفيظ الله سهار	کابل طبي پوهنتون
۶۴	د سينې بطن او حوصلې اناټومي	پوهنوال دوکتور حميدالله حامد	خوست	۶۵	د جهازاتو اناتومي	پوهنوال ډاکتر محمد حسين يار	ننگرهار
۶۶	د عصبي سيستم اناټومي	پوهنیار دوکتور یما صديقي	ننگرهار	۶۷	اناټومي دريم جلد، عصبي سيستم، حواس او اندوکراين غدوات	پوهنمل ډاکتر محمد ناصر نصرتي	ننگرهار
۶۸	د زړه او د وينې د رګونو اناټومي	پوهنمل ډاکتر محمد ناصر نصرتي	ننگرهار	۶۹	د هضمي سيستم اناتومي	پوهنمل ډاکتر محمد ناصر نصرتی	ننگرهار
۷۰	د بولې تناسلي سيستمو اناتومي	پوهنمل ډاکتر محمد ناصر نصرتي	ننگرهار	۷۱	د انسان فزيولوژي او اناټومي	عبدالملک پرهېز	ننگرهار
۷۲	د انسان اناټومي (پورتنی طرف او صدر) ناحيوي او عملي له تسليخ او کلينيک سره	پوهندوی ډاکتر توريالی سهاک	ننگرهار	۷۳	اناتومی و فزيولوژی انسان، جلد دوم	پوهندوی محمد طاهرنسيمی	بلخ
۷۴	اناتومی و فزيولوژی انسان، جلد اول	پوهندوی محمد طاهرنسيمی	بلخ	۷۵	عمومي هستالوژي	پوهندوی ډاکټرفضل الهي	کندهار
۷۶	عمومي هستالوژي	پوهاند ډاکټر خليل احمد بهسودوال	ننگرهار	۷۷	طبی هستولوژي	پوهاند ډاکټر خليل احمد بهسودوال	ننگرهار
۷۸	طبي هستالوژي	پوهاند داکتربری صديقي	خوست	۷۹	هستالوژي	پوهاند ډاکټربری صديقي	کابل طبي پوهنتون
۸۰	د سيستمونو هستالوژي	پوهاند ډاکټر خليل احمد بهسودوال	ننگرهار				
٦. امبريولوژي							
۸۱	عمومي امبريولوژي	پوهاند دوکتوربری صديقي	خوست	۸۲	امبريولوژي	پوهنوال ډاکتر محمد حسين يار	ننگرهار
۸۳	امبريولوژی طبی	پوهندوی داکتربشير نورمل	کابل طبي پوهنتون	۸۴	طبي امبريولوژي	پوهنمل داکتر ناصر نصرتی	ننگرهار
۸۵	امبريولوژی عمومی انسان	پوهندوی داکتربشير نورمل	کابل طبي پوهنتون	۸۶	د انسان عمومي کلينيکي امبريولوژي	پوهنیار ډاکتر عبدالله جان شينواری	ننگرهار
۷. فزيولوژي او پتولوژي							
۸۷	طبي فزيولوژي	داکتر شريف الله	ننگرهار	۸۸	د ځانګړو حسيتونو، پوستکي، اوتونوميک او مرکزي سستم فزيولوژي	پوهنوال دوکتور محب الله شينواری	ننگرهار
۸۹	د اندوکراين، زړه، رګونو او پښتورګو فزيولوژي	پوهنوال دوکتور احسان الله احسان	ننگرهار	۹۰	د تنفسي سيستم فزيولوژي	پوهنوال دوکتور احسان الله احسان	ننگرهار
۹۱	د وينې فزيولوژي	پوهنمل ډاکتر ولي محمد وياړ	کندهار	۹۲	عمومي پتالوژي	پوهاند دوکتور خليل احمد بهسودوال	ننگرهار
۹۳	پتالوژی عمومی	پوهندوی ډاکټرزهرا فروغ	هرات	۹۴	عمومی پتالوژی	پوهندوی دوکتور محمد آصف	ننگرهار
۹۵	د سيستمونو پتالوژي	پوهندوی ډاکترخليل احمد بهسودوال	ننگرهار	۹۶	د سيستمونو پتالوژی دوهمه برخه	پوهاند دوکتور خليل احمد بهسودوال	ننگرهار
۹۷	د قلبي وعايې، وينې، تنفسي او هضمي جهاز پتالوژي	پوهاند دوکتور خليل احمد بهسودوال	ننگرهار	۹۸	د وينې، ججرو، تنفسي جهاز، هضمي جهاز او نويو زېږېدلو فزيولوژي	پوهنوال دوکتور جنت مير مومند	ننگرهار
۹۹	ماليکولي ايمينولوژي	پوهاند ډاکتر خليل احمد بهسودوال	ننگرهار				
۸. مايکروبيولوژي او پرازيتولوژي							
۱۰۰	مايکرو بيولوژی طبی، جلد اول	پوهاند دوکتورعبيدالله عبيد	کابل طبي پوهنتون	۱۰۱	مايکرو بيولوژی طبی، جلد دوم	پوهاند دوکتورعبيدالله عبيد	کابل طبي پوهنتون
۱۰۲	مايکروبيولوژي	پوهاند محمد جمعه حنيف	هرات	۱۰۳	مايکروبيولوژی عمومی	دوکتور شعيب احمد شاخص	هرات
۱۰۴	پرازيتولوژی طبی	پوهاند دوکتورعبيدالله عبيد	کابل طبي پوهنتون	۱۰۵	اساسات پرازيتولوژی طبی	پوهنمل دوکتورمحمد يوسف مبارک	کابل طبي پوهنتون
۱۰۶	د پرازيتولوژی اساسات	ډاکتر محمد صابر	ننگرهار	۱۰۷	طبي پرازيتولوژي	پوهنوال دوکتور غلام جيلاني ولي	ننگرهار
۱۰۸	هلمنتولوژي	پوهنوال ډاکترسيد رفيع الله حليم	ننگرهار				

۹. فارمکولوژي							
۱۰۹	فارمکولوژي	پوهنوال داکتر قمبرعلي حيدري	ننگرهار	۱۱۰	فارمکولوژي، دريم ټوک	پوهنوال سيد قمبر علي حيدري	ننگرهار
۱۱۱	فارمکولوژي، دوهم ټوک	پوهنوال داکتر قمبرعلي حيدري	ننگرهار	۱۱۲	د اتونوم او مرکزي عصبي سيستمونو فارمکولوژي	ډاکترغلام ربی بهسودوال	ننگرهار
۱۱۳	گیاهان طبی مستعمله در تداوی امراض قلبی و وعایی	پوهنوال محمد عثمان بابری	کابل طبي پوهنتون	۱۱۴	امينيو فارمکولوژي	پوهنوال سيد قمبر علي حيدري	ننگرهار
۱۱۵	د درملو د استعمال عملي لارښود (انگليسی/ پښتو)	ډاکتر مالتي ايل وان بلومرودر	خوست	۱۱۶	فارمکولوژي (دريم کال، دوهم سمستر لپاره)	پوهنوال دوکتور غلام ربي بهسودوال	ننگرهار
۱۱۷	د درملو بدې اغيزې	پوهنوال سيد قمبر علي حيدري	ننگرهار				
۱۰. عامه روغتيا							
۱۱۸	د عامې روغتيا اساسات او اداره	پوهنوال ډاکتر محمدعارف رحماني	ننگرهار	۱۱۹	ډيموگرافي او کورنی تنظيم	پوهيالی ډاکتر محمد ابراهيم شيرزی	ننگرهار
۱۲۰	د چاپيريال او أنديزه روغتيا	پوهنوال ډاکتر محمدعارف رحماني	ننگرهار	۱۲۱	تغذيه او روغتيا	پوهيالی ډاکتر محمد هارون	ننگرهار
۱۲۲	تغذيه او سؤتغذيه	پوهنوال داکترعبدالواحد وثيق	قندهار				
۱۱. داخله							
۱۲۳	فزيکل دياگنوزس	پوهنوال ډاکترحفيظ الله اپريدی	ننگرهار	۱۲۴	فزيکی تشخيص او د تاريخچې اخيستنه	پوهاند داکتر شريف الله	ننگرهار
۱۲۵	فزيکي تشخيص	داکتر ناصر جبارخيل	ننگرهار	۱۲۶	فزيکي تشخيص (هادي کلينيکل مېتود)	پوهاند ډاکتر سيف الله هادي	ټول پوهنتونونه
۱۲۷	د داخله ناروغيو تفريقي تشخيص I	پوهاند دوکتور سيف الله هادي	ننگرهار	۱۲۸	د داخله ناروغيو تفريقي تشخيص I	پوهاند دوکتور سيف الله هادي	ننگرهار
۱۲۹	د زړه برقي گراف (ECG)	رنځوروال ډاکتر سيد عبدالله سادات	ننگرهار	۱۳۰	رهنمای عملی مشکلات عام طبی (دری)	داکتر مالتي ال-وان بلومرودر	بلخ
۱۳۱	هيماتولوژي، ايمينولوژي او د ويټامينونو کموالي ناروغۍ	پوهندوی دوکتورايمل شيرزی	ننگرهار	۱۳۲	د طبي عامو ستونځو عملي لارښود (انگليسی)	داکتر مالتي ال-وان بلومرودر	ننگرهار
۱۳۳	د وينې ناروغۍ	پوهاند ډاکتر محمد ظاهرظفرزی	ننگرهار	۱۳۴	د وينې ناروغۍ	پوهنوال دوکتور حيات الله احمدزی	ننگرهار
۱۳۵	اندوکراينولوژي او روماتولوژي	پوهاند داکتر محمد طيب نشاط	ننگرهار	۱۳۶	د پښتورگو ناروغۍ	پوهاند ډاکترمحمد ظاهرظفرزی	ننگرهار
۱۳۷	د هضمي سيستم او پښتوروگو ناروغۍ	پوهندوی ډاکترسيف الله هادي	ننگرهار	۱۳۸	اندوکراينولوژی او روماتولوژي	پوهاند دوکتور سيف الله هادي	ننگرهار
۱۳۹	د خولې او د هضمي سيستم ناروغۍ	پوهاند ډاکترمحمد ظاهرظفرزی	ننگرهار	۱۴۰	د هضمي جهاز ناروغۍ	پوهاند ډاکتر محمد ظاهرظفرزی	ننگرهار
۱۴۱	امراض جهاز هضمی و کبد	دوکتورمحمد يونس فخری	بلخ	۱۴۲	د هضمي جهازاو پښتورگو ناروغۍ	پوهنوال ډاکترعبدالواحد وثيق	کندهار
۱۴۳	د زړه او رگونو ناروغۍ	پوهندوی ډاکتر دل آقا دل	ننگرهار	۱۴۴	د ځيگر ناروغۍ	پوهاند ډاکترمحمد ظاهرظفرزی	ننگرهار
۱۴۵	د تنفسي سيستم او د زړه روماتيزمل ناروغۍ	پوهاند دوکتور سيف الله هادي	ننگرهار	۱۴۶	تنفسي او د زړه د دسامونو روماتيزمل ناروغۍ	پوهاند ډاکترمحمد طيب نشاط	ننگرهار
۱۴۷	د شکرې ناروغي	ډاکتر محمد نعيم همدرد	ننگرهار	۱۴۸	د تنفسي او د زړه روماتيزمل ناروغۍ	پوهندوی داکتر سلام جان شمس	ننگرهار
۱۲. بېړنۍ درملنه							
۱۴۹	بيړنۍ طبي پيښې	پوهنوال داکترعبدالواحد وثيق	قندهار	۱۵۰	د داخلي بيړنې پيښې او د بحران ځانه	پوهنوال ډاکترحفيظ الله اپريدی	ننگرهار
۱۵۱	بيړنۍ درملنې	ډاکتر عبدالولی رنځورمل وردک	خوست	۱۵۲	د بيړنيو پيښو د درملنې لارښود (انگليسی)	پوهنوال ډاکترايمل شيرزی	ننگرها
۱۵۳	بيړنۍ طبي درملنې	داکتر سيد مليار سادات	ننگرهار	۱۵۴	کمک های اوليه	پوهاند دوکتورنجيب الله امرخيل	کابل طبي پوهنتون
۱۳. انکالوژي							
۱۵۵	د سينې سرطان، پېژندنه، درملنه او مخنيوی	پوهنوال ډاکتر نظر محمد سلطانزی ځدراڼ	ننگرهار	۱۵۶	د سرطاني ناروغيو اساسات	پوهاند داکتر محمد ظاهر ظفرزی	ننگرهار
۱۵۷	د وينې سرطان	پوهنوال ډاکتر نظر محمد سلطانزی	ننگرهار	۱۵۸	سرطان او د چاپيريال راديو اکتيويتي	پوهنوال ډاکتر نظر محمد سلطانزی ځدران	ننگرهار
۱۵۹	د سرطاني ناروغيو راډيوتراپي	پوهنوال ډاکتر نظر محمد سلطانزی ځدران	خوست				

۱۴. جراحي

۱۶۰	نرسنگ عملیات خانه	پوهاند دوکتورنجیب الله امرخیل	کابل طبي پوهنتون	۱۶۱	جراحي، د کلینیکي معایناتو سیستم	پوهندوی ډاکټر بادشاه زارعبدالي	خوست
۱۶۲	د عمومي جراحي اساسات	پوهندوی ډاکټر بادشاه زارعبدالي	خوست	۱۶۳	اساسات جراحی	پوهاند ډاکټرنجیب الله امرخیل	کابل طبي پوهنتون
۱۶۴	عمومي جراحي I	پوهندوی ډاکټر بادشاه زارعبدالي	خوست	۱۶۵	عمومي جراحي II	پوهندوی داکتر بادشاه زارعبدالی	خوست
۱۶۶	عمومي جراحي	داکترگل سیما ابراهیم خیل قادري	خوست	۱۶۷	امراض جراحی سیستم هضمی وملحقات آن	پوهاند ډوکتورعبدالوهاب نورا	کابل طبي پوهنتون
۱۶۸	امراض جراحي بطن و ملحقات ان	پوهاند دوکتورمحمد معصوم عزیزی	کابل طبي پوهنتون	۱۶۹	امراض جراحي بطن و ملحقات ان	پوهندوی ډاکټرعبدالخالق دوست	کندهار
۱۷۰	جراحي بطن ، چاپ دوم	پوهاند دوکتورمحمد معصوم عزیزی	کابل طبي پوهنتون	۱۷۱	د ګېډې د ملحقاتو د جراحي ناروغی	پوهنوال دوکتور بادشاه زار عبدالي	خوست
۱۷۲	بطن حاد و مزمن	پوهنوال ډاکټرعبدالغفور ارصاد	هرات	۱۷۳	د پلاستیک جراحي اساسات او تخنیکونه	داکتر الفت هاشمي	ننګرهار
۱۷۴	د کولمو بندش او د پریطوان جراحي ناروغی	پوهاند ډاکټرعبدالرؤف حسان	ننګرهار	۱۷۵	امراض یورولوژی	پوهندوی دوکتورغلام سخی حسنی	کابل طبي پوهنتون
۱۷۶	یورولوژي	پوهندوی دوکتور غازي جمال عبدالناصر	ننګرهار	۱۷۷	یورولوژي	پوهنوال ډاکټرعبدالحد حمید	ننګرهار
۱۷۸	جراحي عصبی	پوهنوال دوکتور عبدالغفور ارصاد	هرات	۱۷۹	عصبي جراحي	پوهندوی دوکتور فضل الرحیم شګیوال	ننګرهار
۱۸۰	عصبي جراحي	پوهندوی داکتر عبدالبصیر منګل	ننګرهار	۱۸۱	عصبي جراحي	پوهاند دکتور بادشاه زار عبدالی	خوست
۱۸۲	د جراحي انکال	رنځوربیار داکتر عجب ګل مومند	ننګرهار	۱۸۳	جراحي عمومی اطفال	پوهنیار داکتر توریالی حکیمی	کابل طبي پوهنتون
۱۸۴	د کوچنیانو جراحي	پوهاند ډاکټر فضل الرحیم شګیوال	ننګرهار	۱۸۵	حاد اپندیسایتیس، تشخیص، اختلاطات او تداوي	پرفیسور دوکتور محمد شریف سروري	ننګرهار
۱۸۶	تروماتولوژی	پوهنوال عبدالغفور ارصاد	هرات	۱۸۷	د صدر ترضیضات	پروفیسور دوکتور محمد شریف سروري	ټول پوهنتونونه

۱۵. ارتوپيدي او انستيزيولوژي

۱۸۸	کسرونه او خلعې	پوهندوی سید بها کریمي	ننګرهار	۱۸۹	اورتوپیدي	پوهندوی ډاکټرسید شال سیدي	ننګرهار
۱۹۰	اورتوپیدي اوکسرونه	پوهنمل ډاکټر محمد همایون مصطفی	کندهار	۱۹۱	د عامو کسرونو ترلې درملنه	پوهندوی دوکتور ظاهر ګل منګل	خوست
۱۹۲	رهنمای انستیزی برای کشورهای رو به انکشاف، جلد اول	دانیل دی موس	کابل طبي پوهنتون	۱۹۳	رهنمای انستیزی برای کشورهای رو به انکشاف، جلد دوم	دانیل دی موس	کابل طبي پوهنتون
۱۹۴	د اورتوپیدي د انتاناتو اساسات، وبلې او د ستون فقرات انتانات	رنځوروال دوکتور سیدالرحمن حکیمي	شیخ زاید				

۱٦. انتاني

۱۹۵	انتاني ناروغی	پوهنوال ډاکټرعبدالناصر جبارخیل	ننګرهار	۱۹۶	امراض انتانی (انگلیسی)	پوهنمل ډاکټر محمد ذکریا امیرزاده	کندهار
۱۹۷	انتاني ناروغی	پوهنوال داکتر حفیظ الله اپریدی	ننګرهار	۱۹۸	د ساري ناروغیو کنټرول	پوهندوی ډاکټر محمد عظیم منګل	ننګرهار
۱۹۹	د کوچنیانو ساري ناروغی	پوهاند دوکتورسلطان محمد صافی	خوست	۲۰۰	د کوچنیانو ساري ناروغی	پوهندوی دوکتور نجیب الله امین	ننګرهار
۲۰۱	امراض ساری اطفال	پوهاند ډاکټرسلطان محمد صافی	کابل طبي پوهنتون	۲۰۲	د ماشومانو انتاني ناروغی	پوهاند دوکتور عبدالستار نیازی	ننګرهار
۲۰۳	توبرکلوز	پوهندوی ډاکټرسید انعام سیدی	ننګرهار	۲۰۴	په ماشومانو کې نری رنځ	پوهنمل ډاکټر حقیق الله چاردیوال	ننګرهار
۲۰۵	د توبرکلوز ناورغی	ډاکټرمحمد ناصرناصري	کندهار	۲۰۶	د سینې ناروغی او توبرکلوز	ډاکټر ناصر محمد شینواری	کندهار
۲۰۷	ملاریا	دوکتور محمد اسحاق شریفي	ننګرهار	۲۰۸	سارس – ۲ او کووید- 19	پروفیسور دوکتور محمد شریف سروري	ټول پوهنتونونه
۲۰۹	د ځیګر ویروسي التهاب (طبي تشخیص او درملنه)	دوکتور محمد اسحاق شریفي	ننګرهار	۲۱۰	شل خپرې ساري ناروغی	ډاکټر غلام سرور ظهیر	شیخ زاید

۱۷. اطفال							
۲۱۱	اطفال	پوهنوال ډاکټرمحمد رسول فضلي	ننگرهار	۲۱۲	د کوچنیانو ناروغی I	پوهنوال دوکتور عبدالستار نیازی	ننگرهار
۲۱۳	د کوچنیانو ناروغی II	پوهنوال دوکتور عبدالستار نیازی	ننگرهار	۲۱۴	د ماشومانو کلینکی معاینات	پوهنوال ډاکټرناصرکاموال	ننگرهار
۲۱۵	روش های ارزیابی کلنیکی اطفال	پوهندوی ډاکټرفاروق حمیدي	کابل طبي پوهنتون	۲۱۶	د کوچنیانو ناروغی تکست بوک	پوهاند ډاکټر سلطان محمد صافی	کابل طبي پوهنتون
۲۱۷	د کوچنیانو د درملنX لاربنود (انگلیسی)	پوهندوی داکترمنصوراسلمزی	ننگرهار	۲۱۸	د کوچنیانو تغذیه	پوهنمل ډاکټر نجیب الله امین	ننگرهار
۲۱۹	د کوچنیانو خوارځواکی	پوهندوی ډاکټرسمیع الله حیات	ننگرهار	۲۲۰	د کوچنیانو د وینې ناروغی	پوهندوی داکتر منصور اسلمزی	ننگرهار
۲۲۱	د نوي زیږیدلي ماشوم څارنه	پوهندوی ډاکټر ناصر خان کامه وال	ننگرهار	۲۲۲	د نیونتالوژی او کوچنیانود ناروغیو کلینیکی هندبوک	پوهندوی داکتر منصور اسلمزی	ننگرهار
۲۲۳	د ماشومانو د ناروغیو عملي لاربنود	داکتر مالتې ال-وان بلومرودر	ننگرهار	۲۲۴	نیونتولوژي	پوهنوال داکتر عبدالستار نیازی	ننگرهار
۲۲۵	د کوچنیانو د جهازاتو معمولې ناروغی I	پوهنوال داکتر عبدالستار نیازی	ننگرهار	۲۲۶	د کوچنیانو نارغی د پنځم صف لپاره لومړی سمیستر	پوهنوال داکتر عبدالستار نیازی	ننگرهار
۲۲۷	د ماشومانو د معدې معایي سیستم او ینې ناروغی	پوهنمل ډاکټر ولي گل مخلص	خوست	۲۲۸	د کوچنیانو د جهازاتو معمولې ناروغی II	پوهنوال داکتر عبدالستار نیازی	ننگرهار
۲۲۹	د کوچنیانو ناروغی I	پوهاند دوکتور احمد سیر احمدي	ننگرهار	۲۳۰	د کوچنیانو ناروغی II	پوهاند دوکتور احمد سیر احمدي	ننگرهار
۲۳۱	د ماشومانو تنفسي، زړه، وینې او پښتورگې ناروغی	پوهاند داکتر نجیب الله امین	ننگرهار	۲۳۲	د کوچنیانو ناروغی	پوهاند ډاکټر سلطان محمد صافی	کابل طبي پوهنتون
۲۳۳	د ماشومانو اساسات، هضمي، اندوکراین او عصبي ناروغی	پوهاند ډاکټر سمیع الله حیات	ننگرهار	۲۳۴	معاینات کلینیکی اطفال بطور ساده	پوهندوی دوکتور سید نجم الدین جلال	کابل طبي پوهنتون
۱۸. ولادي/ نسایي							
۲۳۵	زیږون	پوهنمل ډاکټرمریم اکرم معصوم	ننگرهار	۲۳۶	ولادی جراحی، لمړی ټوک	داکترعجب گل مومند	قندهار
۲۳۷	کتاب ولادی	پوهندوی داکترحسن فرید	هرات	۲۳۸	ولادی جراحی، دوهم ټوک	داکترعجب گل مومند	قندهار
۲۳۹	امراض نسایی	پوهندوی ډاکټر حسن فرید	هرات	۲۴۰	د ثدیې ناروغی	پروفیسور دوکتور محمد شریف سروري	ټول پوهنتونونه
۲۴۱	نسایي ناروغی	پوهندوی دوکتورس تورپیکې اپریدی	ننگرهار	۲۴۲	امېندواري او زیږون	پوهنوال دوکتورس حفیظه سهاک	ننگرهار
۱۹. روانشناسي، رواني/عقلي اوعصبي							
۲۴۳	امراض روانی I	پوهندوی دوکتورعبدالعزیز نادری	کابل طبي پوهنتون	۲۴۴	امراض روانی II	پوهندوی دوکتورعبدالعزیز نادری	کابل طبي پوهنتون
۲۴۵	رواني رنځپوهنه	پوهندوی ډاکټرجهان شاه تنی	خوست	۲۴۶	نشه يي توکي او اړونده ناروغی	ډاکټر محمد سمین ستانکزی	ننگرهار
۲۴۷	د رواني روغتیایي ستونځو عملي لاربنود (انگلیسي)	سیان نیکولاس	ننگرهار	۲۴۸	عصبي ناروغی	پوهنمل داکتر بلال پاینده	ننگرهار
۲۴۹	عقلي ناروغی	پوهنمل داکتر بلال پاینده	ننگرهار	۲۵۰	روانشناسی و ضرورت آن در جامعهٔ افغانستان	داکتر اعظم دادفر	کابل
۲۵۱	روانشناسی عمومی	پوهاند ماریا صاعد سلطانی	بلخ	۲۵۲	عصبي معاینات او سلوکپوهنه	پوهنوال دوکتور جهان شاه تنی	خوست
۲۰. راديولوژي/ تصویربرداری طبی							
۲۵۳	کلینکي رادیولوژي	پوهنوال ډاکټر غلام سخي رحمانزی	ننگرهار	۲۵۴	د زړه او سږو د ناروغیو تشخیصیه رادیولوژي	پوهنیار ډاکټر شاه محمد رنځورمل	خوست
۲۵۵	تشخیصي رادیولوژي	پوهنوال ډاکټر غلام سخي رحمانزی	ننگرهار	۲۵۶	تصویرې یا ترسیمې رادیولوژی	پوهنوال ډاکټر غلام سخي رحمانزی	ننگرهار
۲۵۷	رادیولوژي ازموینې او ناروغتیاوې	پوهنوال ډاکټر غلام سخي رحمانزی	ننگرهار	۲۵۸	التراساوند تشخیصیه	پوهندوی ډاکټر محمد نواب کمال	کابل طبي پوهنتون
۲۵۹	رادیولوژي	پوهنوال ډاکټر سید عارف ویاړ	ننگرهار	۲۶۰	الټراسونډ	ډاکټر محمد یونس سلطانی	ننگرهار
۲۶۱	تشخیصیه رادیولوژي، سینه یا صدر، دوهمه برخه	پوهنوال دوکتور نور محمد شینواری	ننگرهار	۲۶۲	تلویزیوني آزموینې	پوهندوی ډاکټر نجیب الله خلیلي	ننگرهار
۲۶۳	د الټراسونډ طبي کارول	پوهنوال ډاکټر نظر محمد سلطانزی ځدران					
۲۱. چشم، گوش و گلو، جلدی							
۲۶۴	د پوستکي ناروغی	پوهندوی دوکتوراسدالله شینواری	ننگرهار	۲۶۵	دغوږ، پزې او ستوني ناروغی	داکتر عزیز الله فقیر	کندهار
۲۶۶	د سترگو ناروغی	پوهنمل داکترخالد یار	ننگرهار	۲۶۷	دغوږ، پزې او ستوني ناروغی	پوهنمل دوکتور میر محمد اسحاق خاورین	ننگرهار

۲۶۸	د پوستکي ناروغی	پوهنمل ډاکتر سید انور اکبري	ننګرهار	۲۶۹	د سترګې کلینیکي ناروغی	پوهنوال ډاکتر عبدالبصیر صافی	ننګرهار

۲۲. غاښونه

۲۷۰	رهنمای کلینیکی برای داکتران دندان	داکتر سید معروف سیرت	ټول پوهنتونونه				

۲۳. انجنیري

۲۷۱	د اوبو رسولو انجنیري	پروفیسور انجنیر محمد عیسی تنها	ننګرهار	۲۷۲	د فاضله اوبو انجنیري	پوهاند انجنیر زلمی خالقي	ننګرهار
۲۷۳	چګونګی مصرف انرژی در ساختمان های رهایشی	دوکتور انجنیر محمد عمر تیموری	ننګرهار	۲۷۴	تأسیسات و تجهیزات تخنیکی ساختمان	دوکتور انجنیر محمد عمر تیموری	پولی تخنیک کابل
۲۷۵	د ساختمانونو تحلیل، لومړی برخه	پوهاند محمد اسحق رازقي	ننګرهار	۲۷۶	د ساختمانونو تحلیل، دوهمه برخه	پوهاند محمد اسحق رازقي	ننګرهار
۲۷۷	د مهندسانو د پاره ساختماني ستاتیک زده کړه	دیپلوم انجنیر اسدالله ملکزی	ننګرهار	۲۷۸	د جوړښتونو تحلیل، لومړی برخه	پروفیسور حفیظ الله وردک او پروفیسور دکتور زرجان بها	خوست
۲۷۹	د جوړښتونو تحلیل، دوهمه برخه	پروفیسور حفیظ الله وردک او پروفیسور دکتور زرجان بها	خوست	۲۸۰	۴۵ انجنیري درسي کتابونه (DVD)	ټول پوهنتونونه	ټول پوهنتونونه
۲۸۱	د موادو مقاومت	پوهنمل بهرام امیري	خوست	۲۸۲	اوسپنیز کانکریټي عناصر I	پوهنوال دیپلوم انجنیر عبادالرحمن مومند	ننګرهار
۲۸۳	اوسپنیز کانکریټي عناصر ډیزاین دوهمه برخه، لومړی ټوک	پوهاند دیپلوم انجنیر عبادالرحمن مومند	ننګرهار	۲۸۴	اوسپنیز کانکریټي عناصر ډیزاین دوهمه برخه، دوهم ټوک	پوهاند دیپلوم انجنیر عبادالرحمن مومند	ننګرهار
۲۸۵	د اوسپیز کانکریټي عناصرو د لومړی صنفي کار مېتودیکي لارښود	پوهندوی انجنیر عبادالرحمن مومند	ننګرهار	۲۸۶	د جامداتو میخانیک	پوهنوال محمد اسحق رازقي	ننګرهار
۲۸۷	په سیول انجنیري کې د اټوکډ استعمال	پوهنوال میا پاچا میاخېل	ننګرهار	۲۸۸	د سرخلاصو کانالونو هایدرولیک	پوهنوال میا پاچا میاخېل	ننګرهار
۲۸۹	د لوبو لارو د هندسي عناصرو ډیزاین	پوهنیار انجنیر م. شاکر فاروقي	ننګرهار	۲۹۰	د ودانیو د تودولو تخنیک، لومړی برخه، د سون تخنیک	ډاکتر غلام فاروق میر احمدي	ننګرهار
۲۹۱	د تهداب انجنیري	پوهاند انجنیر زلمی خالقي	ننګرهار	۲۹۲	معیارهای جدید اعمار ساختمان	دوکتور انجنیر محمد عمر تیموری	ننګرهار
۲۹۳	د انجنیري میخانیک	پوهنوال محمد اسحق رازقي	ننګرهار	۲۹۴	عمومي تخنیکي رسم	پوهیالی فضل اکبر	ننګرهار
۲۹۵	انرژي سپما کوونکې ودانی	انجنیر اسد الله ملکزی	ننګرهار	۲۹۶	انجنیري جیودوزي (سروې)	پوهندی ګل حکیم شاه سیدي	ننګرهار
۲۹۷	د ساختمان د جوړولو طریقې I	پوهاند انجنیر محمد عیسی تنها	ننګرهار	۲۹۸	رهنمود موثریت حفظ انرژی در تعمیرات	داکتر انجنیر محمد عمر تیموری	کابل
۲۹۹	اعمار ساختمانها (اساسات، مواد و سیستم ها)	پوهندوی انجنیر امان الله فقیری	کابل پولیتخنیک	۳۰۰	د ساختمان د جوړولو طریقې II	پوهاند انجنیر محمد عیسی تنها	ننګرهار
۳۰۱	د ودانیو د جوړولو مهندسي اساسات II	دیپلوم انجنیر اسدالله ملکزی	ننګرهار	۳۰۲	کید او ګرافیک	پوهنوال دیپلوم انجنیر بهاوالدین جلالي	ننګرهار
۳۰۳	د اوبو لګولو انجنیري	پوهندوی دیپلوم انجنیر اصغر غفورزی	ننګرهار	۳۰۴	د ودانیو د جوړولو مهندسي اساسات، لومړی ټوک	دیپلوم انجنیر اسدالله ملکزی	ننګرهار
۳۰۵	د جوړښتونو تحلیل، دریيمه برخه	پروفیسور حفیظ الله وردک او پروفیسور دکتور زرجان بها	خوست	۳۰۶	اساسات هندسه ترسیمی مسطح	پوهنوال سید یوسف مانووال	بلخ
۳۰۷	د پولادي عناصرو ډیزاین لومړی ټوک	محمد ذکریا محمدي	ننګرهار	۳۰۸	د پولادي عناصرو ډیزاین دوهم ټوک	محمد ذکریا محمدي	ننګرهار

۲۴. زراعت

۳۰۹	د خاورې تخریب او د چاپېریال ککړتیا	پوهنیار محمد حنیف هاشمي	خوست	۳۱۰	د کرنیزو محصولاتو بازار موندنه	پوهاند محمد طیب	ننګرهار
۳۱۱	د کرنې تشریحي قاموس، انګلیسی -پښتو	پوهاند محمد بشیر دودیال	ننګرهار	۳۱۲	د کورنیو الوتونکو د روزنې بنسټونه	پوهاند میر حاتم نیازی	ننګرهار
۳۱۳	نیماتولوژي	پوهنوال حسین آرمان	ننګرهار	۳۱۴	نباتي فزیولوژي لومړی جلد	پوهنمل محمد طاهر میاخېل	خوست
۳۱۵	نباتي فزیولوژي، دوهم جلد	پوهنمل محمد طاهر میاخېل	خوست	۳۱۶	عمومي نباتات	پوهنوال عبدالخلیل افغاني	شیخ زاید

۲۵. وترنري

۳۱۷	وترنري عمومي پتالوژي	پوهندوی محمد طاهر کاکړ	ننګرهار	۳۱۸	حیواني تغذیه، لومړی برخه	پوهندوی روزي خان صادق	ننګرهار
۳۱۹	حیواني تغذیه، دوهمه برخه	پوهندوی روزي خان صادق	ننګرهار	۳۲۰	وترنري داخله	پوهنوال پیر محمد ستانکزی	ننګرهار
۳۲۱	وترنري فارمکولوژي دوهمه برخه	پوهنوال محمد بایر درمل	ننګرهار	۳۲۲	د ژوبو فزیولوژي	پوهاند غنچه ګل حبیب صافی	ننګرهار

۲۶. ژورنالېزم

۳۲۳	د راډیويي خپرونو تولید	پوهنوال دوکتور ماستر واحدي	خوست	۳۲۴	د ټلویزیوني خپرونو تولید	پوهنوال ډاکتر ماستر واحدي	خوست
۳۲۵	اطلاعاتو ته د لاسرسي لارې چارې	دانش کروخېل	ننګرهار				

۲۷. اقتصاد او مدېریت

۳۲۶	د اقتصادي پرمختیا تیوري	پوهاند محمد بشیر دویال	ننګرهار	۳۲۷	د اقتصاد او تجارت اصطلاحات (انګلیسي - پښتو تشریحي قاموس)	پوهنیار عبدالله عادل او امان الله ورین	ننګرهار
۳۲۸	تیوری و سیاست بودجه عامه	پوهنوال داکتر سید محمد تینګار	کابل	۳۲۹	د پروژې مدیریت په عمل کې	محمد داود علم او یو اف. ګهل	ننګرهار
۳۳۰	د پروژې تحلیل او مدیریت	پوهاند محمد بشیر دویال	ننګرهار	۳۳۱	مبادی اقتصاد زراعتی	پوهاند ولی محمد فائز	بلخ
۳۳۲	صنعتي اقتصاد	پوهاند محمد بشیر دودیال	ننګرهار	۳۳۳	د اقتصاد د علم اساسات	شېر خان حساس	ننګرهار

۳۳۴	مرکزي بانک او پرمختللي پولي سیاستونه	پوهاند دوکتور عبدالقیوم عارف	خوست	۳۳۵	اقتصادي جیولوجي (کانپوهنه-فلزي کانونه)	پوهاند دوکتور شریف الله سهاک	ننگرهار
۳۳۶	عامه اقتصاد	پوهندوی ریحان الله رحیمي	ننگرهار	۳۳۷	احصاییه	پوهاند محمد بشیر دودیال	ننگرهار
۳۳۸	د احصایې اساسات	پوهنیارمحمد اغا ضیاء	کندهار	۳۳۹	د اقتصاد تشریحي قاموس انکلیسي ـ پښتو	پوهاند محمد بشیر دودیال	ننگرهار
۳۴۰	د اداري اومدیریت تشریحي قاموس انگلیسي ـ پښتو	پوهاند محمد بشیر دودیال	ننگرهار				
۲۸. عامه اداره او پالیسي							
۳۴۱	د څېړنې میتودولوژي	پوهنیار نثار احمد مصلح	ننگرهار	۳۴۲	رهبري له تیوري تر عمله	پوهنمل محمد عرفان قریشي	ننگرهار
۳۴۳	د سازماني اړیکو مدیریت	پوهاند محمد بشیر دودیال	ننگرهار	۳۴۴	نړیوالې تړونې	احسان الله آرینزی	ننگرهار
۳۴۵	د بشري سرچینو مدیریت	پوهنمل منصور فقیرزی	ننگرهار	۳۴۶	پېداګوژي	پوهنیار راز محمد فیضي	ننگرهار
۳۴۷	گروههای اجتماعی بسته (مطالعه جامعه شناختی سکتها)	داکتر احمد سیر مهجور	کابل پوهنتون	۳۴۸	د رهبرۍ اصول	پوهنمل محمد عرغان قریشي	ننگرهار
۳۴۹	د بشري سرچینو د مدیریت اړین توکي	پوهندوی نعیم جان سروري	ننگرهار				
۲۹. چاپېریال او جغرافیه							
۳۵۰	د نفوسو جغرافیه	پوهنوال لطف الله صافی	ننگرهار	۳۵۱	حیاتي جغرافیه	پوهاند لطف الله صافی	ننگرهار
۳۵۲	جیومورفولوژي	پوهنوال عزت الله	ننگرهار	۳۵۳	اقلیم پوهنه	پوهاند عزت الله سایل	ننگرهار
۳۵۴	کارتو گرافی با اساسات توپوگرافی	پوهنوال دوکتور محمد طاهر عنایت	ننگرهار	۳۵۵	د میتیورولوژي مبادي	پوهنوال عبدالغیاث صافی	ننگرهار
۳۵۶	د ژوند چاپېریال	پوهاند عارف الله مندوزی	ننگرهار	۳۵۷	گرم شدن کره زمین	محمد نعیم نسین	بلخ
۳۰. ریاضیات							
۳۵۸	عمومي ریاضیات	پوهنوال گل محمد جنت زی	خوست	۳۵۹	د عالي ریاضیاتو عمومي کورس	پوهندوی محب الرحمن جنتي	ننگرهار
۳۶۰	عالي کلکولس I، 434 A ریاضي	پوهندوی حمیدالله یار	ننگرهار	۳۶۱	عالي کلکولس II	پوهندوی نظر محمد	ننگرهار
۳۶۲	الجبر او د عددونو تیوري، لومړی برخه	سطان احمد نیازمن	ننگرهار	۳۶۳	خطي الجبر	داکتر عبدالله مهمند	ننگرهار
۳۶۴	کلکولس او تحلیلي هندسه I	پوهندوی سید شیر آقا سیدي	ننگرهار	۳۶۵	کلکولس او تحلیلي هندسه II	پوهندوی سید شیر آقا سیدي	ننگرهار
۳۶۶	الجبر او د عددونو تیوري، دوهمه برخه	سلطان احمد نیازمن	ننگرهار	۳۶۷	د ریاضي په هکله خبرې اترې	سلطان احمد نیازمن	ننگرهار
۳۶۸	الجبر معاصر	داکتر عبدالله مهمند	بلخ	۳۶۹	معاصر الجبر	ډاکتر عبدالله مهمند	خوست
۳۷۰	سېټونه او هر څه د هغوی په هکله	لیف بوکوفسکي/ سلطان احمد نیازمن	ننگرهار	۳۷۱	د ریاضي منطق	سطان احمد نیازمن	ننگرهار
۳۷۲	د انجنیري اساسي ریاضي I	پوهندوی عبدالغفور نیازی	ننگرهار	۳۷۳	د انجنیري اساسي ریاضي II	پوهندوی عبدالغفور نیازی	ننگرهار
۳۷۴	د تحلیلي هندسې I	سید شیر اقا سیدي	ننگرهار	۳۷۵	انالیزریاضی I	سید یوسف مانووال	بلخ
۳۷۶	عالي ریاضي د تسریحي مثالونو سره	ډاکتر عبدالله وردک	شیخ زاید				
۳۱. ژبه او ادبیات							
۳۷۷	آلماني د افغانانو لپاره	ډاکتر یحیی وردک	ببلابېل	۳۷۸	آلمانی برای افغانها به دری	ډاکتر یحیی وردک	ببلابېل
۳۷۹	د جرمني ژبې آسانه زده کړه، له اساساتو نه تر ادبیاتو پورې	ډاکتر اکرم ملکزی	ننگرهار	۳۸۰	د افغانستان د پوهنتونونو د درسي کتابونو چاپول (پښتو)	ډاکتر یحیی وردک	ټولو ته
۳۸۱	د افغانستان د پوهنتونونو د درسي کتابونو چاپول (انگلیسي)	ډاکتر یحیی وردک	ټولو ته	۳۸۲	د کتاب خپرولو لنډ لارښود	ډاکتر یحیی وردک	ټولو ته
۳۸۳	د کتاب خپرولو لنډ لارښود (انگلیسي)	ډاکتر یحیی وردک	ټولو ته	۳۸۴	جرمني ـ پښتو ستر قاموس (لومړی ټوک)	ډاکتر اکرم ملکزی	شیخ زاید پوهنتون
۳۸۵	پښتو ـ انگلیسي قاموس	رحیمزی	ننگرهار	۳۸۶	جرمني ـ پښتو ستر قاموس (دوهم ټوک)	ډاکتر اکرم ملکزی	شیخ زاید پوهنتون
۳۲. کمپیوټرساینس							
۳۸۷	د ډېټابېس اساسات	زرگی حبیبي	ننگرهار	۳۸۸	د کمپیوټرو جال	سلطان احمد نیازمن	ننگرهار
۳۸۹	د کمپیوټر جوړښت او اسمبلي ژبه	پوهندوی بادام نیازی	ننگرهار				

مرسته کوونکي: (Kinderhilfe-Afghanistan (x۲۵۰)، DAAD (x۹۶)، Konrad Adenauer Stiftung (x ۸) ، DAUG (x7)، Michael Klett(x۶)، Afghanistan-Schulen (x4)،

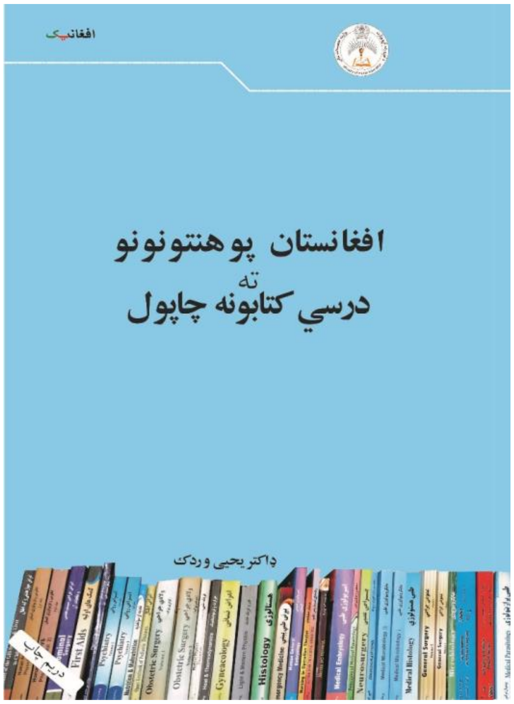
سرکنسولگری جمهوری فدرال آلمان مزارشریف (x۱)، inasys (x۱)، humedica(x۲)، ، SlovakAid (x۲)، (x۲)، صافي بنسټ (x۱) او افغانیک

تطبیق کوونکی: ډاکتر یحیی وردک، د لوړو زده کړو وزارت، څلورمه کارته، کابل افغانستان، مې ۲۰۲۳

موبایل:0780232310، ۰۷۰۷۳۲۰۸۴۴، ایمیل: info@ecampus-afghanistan.org ,www.mohe.gov.af

ټول کتابونه له دې وېب پاڼو څخه ډونلوډولای شئ: www.ecampus-afghanistan.org

افغاني درسي کتابونو ته آنلاین لاس رسی
Access to Online Afghan Textbooks
ecampus-Afghanistan.org
Full version of all textbooks can be downloaded as PDF from above website.
if you want to publish your textbooks please contact us: Dr. Yahya Wardak, Ministry of Higher Education, Kabul, Office: 0706320844, Email: info@ecampus-afghanistan.org



افغانیک
افغانستان پوهنتونونو
ته
درسي کتابونه چاپول
ډاکتر یحیی وردک

افغانیک
د کتاب خپرولو
لنډ لارښود
فزیکي تشخیص
داکتر یحیی وردک

Publishing Textbooks

Honorable lecturers and dear students!

The lack of quality textbooks in the universities of Afghanistan is a serious issue, which is repeatedly challenging students and teachers alike. To tackle this issue, we have initiated the process of providing textbooks to the students of Afghan universities .

For this reason, we have published 389 different textbooks of Medicine, Engineering, Science, Economics, Journalism, and Agriculture from Nangarhar, Khost, Kandahar, Herat, Balkh, Al-Beroni, Kabul, Kabul Polytechnic, and Kabul Medical universities since 2010. The book you are holding in your hands is a sample of a printed textbook. It should be mentioned that all these books have been distributed among all Afghan universities and many other institutions and organizations for free. All the published textbooks can be downloaded from www.ecampus-afghanistan.org .

The Afghan National Higher Education Strategy (2010-2014) states: *"Funds will be made available to encourage the writing and publication of textbooks in Dari and Pashto. Especially in priority areas, to improve the quality of teaching and learning and give students access to state–of–the–art information. In the meantime, translation of English language textbooks and journals into Dari and Pashto is a major challenge for curriculum reform. Without this facility it would not be possible for university students and faculty to access modern developments as knowledge in all disciplines accumulates at a rapid and exponential pace, in particular this is a huge obstacle for establishing a research culture. The Ministry of Higher Education together with the universities will examine strategies to overcome this deficit ".*

We would like to continue this project and to end the method of manual notes and papers. Based on the request of higher education institutions, there is the need to publish about 100 different textbooks each year.

I would like to ask all the lecturers to write new textbooks, translate or revise their lecture notes or written books and share them with us to be published. We will ensure quality composition, printing and distribution to Afghan universities free of charge. I would like the students to encourage and assist their lecturers in this regard. We welcome any recommendations and suggestions for improvement.

It is worth mentioning that the authors and publishers tried to prepare the books according to the international standards, but if there is any problem in the book, we kindly request the readers to send their comments to us or the authors in order to be corrected for future revised editions.

We are very thankful to Kinderhilfe-Afghanistan (German Aid for Afghan Children) and its director Dr. Eroes, who has provided fund for this book. We would also like to mention that he has provided funds for 250 medical and non-medical textbooks so far.
I would like to cordially thank Chancellor of Universities, Deans of faculties, and lecturers for their continuous cooperation and support for this project .
I am also thankful to all those lecturers who encouraged us and gave us all these books to be published and distributed all over Afghanistan. Finally, I would like to express my appreciation for the efforts of my colleagues Hekmatullah Aziz, Fahim Habibi, Gul Agha Ahmadi and Hewad Safi in the office for publishing and distributing the textbooks.

Dr. Yahya Wardak
Ministry of Higher Education, Kabul, Afghanistan, June, 2023
Mobile: 0706320844, 0780232310
Email: info@ecampus-afghanistan.org

Book Name	Computer Architecture & Assembly Language
Translator	Assist Prof Badam Niazai
Publisher	Nangarhar University, Faculty of Computer Science
Website	www.nu.edu.af
Published	2023, First Edition
Copies	1000
Serial No	384
Download	www.ecampus-afghanistan.org

This publication was financed by **Kinderhilfe-Afghanistan** (German Aid for Afghan Children) a private initiative of the Eroes family in Germany.

Administrative and technical support by Afghanic.

The contents and textual structure of this book have been developed by concerning translator and relevant faculty and being responsible for it.

Funding and supporting agencies are not holding any responsibilities.

If you want to publish your textbooks, please contact us:
Dr. Yahya Wardak, Ministry of Higher Education, Karte – 4, Kabul
Office 0780232310, 0706320844
Email info@ecampus-afghanistan.org

All rights reserved with the translator.

Printed in Afghanistan 2023

ISBN 978-9936-622-65-4