Қазақстан Республикасы Білім және ғылым министрлігі

«Гумарбек Даукеев атындағы Алматы энергетика және
байланыс университеті»
коммерциялық емес акционерлік қоғамының

Оқу құралы

Цифрлық схемотехника

Мусапирова Г.Д.

Алматы
2022

Кіріспе

Заманауи цифрлық жүйелердің құрылымында өзара байланысты және өзара тәуелді екі фактор ескеріледі:

– жүйенің аппаратық бөлігі (hardware) қойылған мәселелердің іс жүзінде орындалуын қамтамасыз ету керек;

– енгізілген командалар жүйесі (software) жобаланған жүйенің жеке бөліктемелерінің жұмысы мен олардың өзара әрекеттесуін шешілетін мәселелерге сәйкесті бағдарлауға мүмкіндік беру керек.

Осы факторлар ескеріле отырып құрылған, ұсынылған оқу құралы цифрлық жүйелердің аппаратық құрылым принциптерін де, қойылған мәселелерді шешуге бағытталған әртүрлі бағдарламалық құрылымдардың жүзеге асырылу принциптерін де игеруге мүмкіндік береді.

Оқу құралының бірінші және екінші тарауында цифрлық құрылғылардың құрылым негізін қалаушы әртүрлі қиыстырма және тізбектеме құрылғылардың құрылым принциптері мен жұмыс тәртібі және олардың ерекше пайдаланылатын іс жүзіндегі мысалдары келтірілген. Жады құрылғыларының әртүрлі құрылым принциптері оқу құралының үшінші тарауында қарастырылады.

Микропроцессорлық жүйелердің негізгі құрылым принциптері мен олардың жұмысының ұйымдастырылу тәртібі оқу құралының төртінші және бесінші тарауында қарастырылған. Қарастыруға нақтылы мысал ретінде, бастапқы оқуға аса ыңғайлы келетін, сегізразрядты Intel 8085 микропроцессоры алынды. Заманауи микропроцессорлардың арғы атасы саналатын, қарапайым Intel 8085 микропроцессорының осы уақытқа дейін әртүрлі басқару жүйелерін құруға негіз құратынын да айта кету керек.

Оқу құралының алтыншы тарауы микропроцессорлық жүйелердің жалпы құрылым принциптерінің іс жүзінде жүзеге асырылу мүмкіндігін суреттеу мақсатында бағдарламалы Electronics Workbench модельдеу жүйесінде құрылған микропроцессорлық жүйенің моделі, оның құрама буындары мен ерекше командалар жүйесі және сәйкесті бағдарламалы құрылымдарының жүзеге асырылу жолдары қарастырылады.

Ұсынылған оқу құралы автордың көпжылдық педагогикалық қызметінің негізінде құрастырылды және ол 5В070400 – Есептеу техникасы және бағдарламалық қамтамасыз ету мамандығында оқитын студенттерге арналған.

1 Цифрлық жүйелердің құрылым негіздері

Цифрлық құрылғылар істеу тәртібіне қарай екі түрге бөлінеді, олар – қиыстырма құрылғылар және тізбектеме құрылғылар.

Қиыстырма құрылғылардың шығыстарының жағдайы, яғни олардың сәйкесті шықпаларындағы сигнал деңгейлері, олардың кірістеріндегі осы мезеттегі сигнал деңгейлерімен ғана анықталады да, осы кірістерде бұрын болған сигналдарға тәуелді болмайды. Сондықтан бұндай құрылғыларды жадысыз құрылғылар деп атауға болады.

Тізбектеме құрылғылардың шығыстарының жағдайы олардың кірістеріндегі осы мезеттегі сигнал деңгейлерімен қатар осы құрылғылардың кірістерінде алдыңғы мезеттерде болған сигналдарға байланысты туған олардың ішкі жағдайымен де анықталады. Сондықтан бұндай құрылғыларды жадылы құрылғылар деп атауға болады.

1.1 Логикалық функциялар мен логикалық элементтер

Цифрлық құрылғылардың (қиыстырма құрылғылардың да, тізбектеме құрылғылардың да) құрылым негізін логикалық элементтер құрады.

Логикалық элементтер – логикалық функцияларды жүзеге асыруға арналған құрылғылар. Логикалық функциялар әдетте екі түрлі тәсілмен беріледі: олар – аналитикалық және кестелік тәсілдер.

Кестелік тәсілдің пайдаланылуы кезінде аргумент мәндерінің барлық болар қиылыстары мен оларға сәйкесті логикалық функция мәндері кесте түрінде беріледі (бұл кесте қарастырылушы функцияның ақиқаттық кестесі деп аталады). Функцияның аргумент саны n болса, онда аргумент мәндерінің барлық қиылыстарының (әдетте олар аргумент мәндерінің жинақтары деп аталады) саны 2^n болады. Мысалы, аргумент мәндерінің жинақтарының саны $n = 2$ кезінде $2^2 = 4$, ал $n = 3$ кезінде $2^3 = 8$ болады.

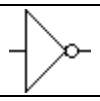
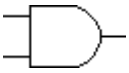
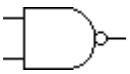
Логикалық функцияның аналитикалық тәсілмен берілуінде функция логикалық өрнекпен көрсетіледі де, бұл өрнек функция аргументтеріне жүргізілетін логикалық операциялар мен олардың орындалу тәртібін тиянақтайды.

1.1-кестеде негізгі логикалық функциялардың аталымдары мен өрнектері және осы логикалық функцияларды жүзеге асырушы сәйкесті элементтердің шартты сызба белгілемелері көрсетілген.

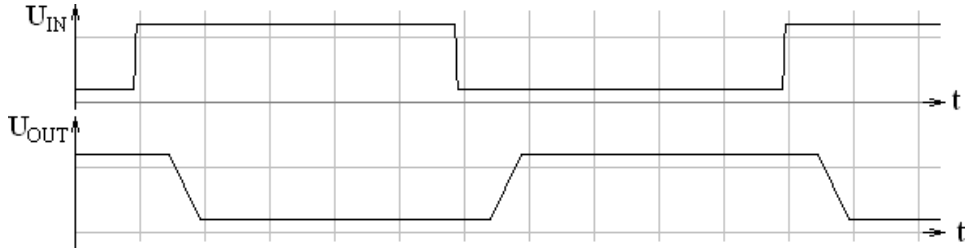
Кейбір элементтерді сәйкесті ақиқаттық кестелеріне қарап, олардың ерекше қолданылым мүмкіндіктерін байқауға болады. Мысалы, екікірісті AND элементінің кірістерінің біреуінде (оны басқару кірісі деп аталық) нөлдік деңгей болғанда элемент шығысы оның екінші кірісіндегі сигнал деңгейіне тәуелсіз болады да, ол әрқашан нөлдік деңгейде болады. Элементтің басқару кірісіне бірлік деңгей берілгенде оның екінші кірісіндегі (оны ақпарат кірісі деп аталық) сигналды элемент шығысына қарай өзгеріссіз өткізеді. Демек, бұндай элементті ақпараттық сигналды элемент шығысына қосылған

құрылғыға, басқару сигналының деңгейіне байланысты, өткізу-өткізбеу қызметін атқаратын кілт ретінде пайдалануға болады. Осы сияқты, XOR элементінің кірістерінің біреуінде (басқару кірісі) нөлдік деңгей болғанда оның екінші кірісіндегі (ақпарат кірісі) сигнал элемент арқылы өзгеріссіз өтеді де, ал оның басқару кірісіне бірлік деңгей берілгенде ақпараттық сигнал элемент арқылы терістеліп өтеді.

1.1 к е с т е

		NOT	OR	AND	NOR	NAND	XOR	XNOR
*X*1	*X*0	$\overline{0}$	$1 \vee 0$	$1 \wedge 0$	$\overline{1 \vee 0}$	$\overline{1 \wedge 0}$	$1 \oplus 0$	$\overline{1 \oplus 0}$
								
0	0	1	0	0	1	1	0	1
0	1	0	1	0	0	0	1	0
1	0	1	1	0	0	0	1	0
1	1	0	1	1	0	0	0	1

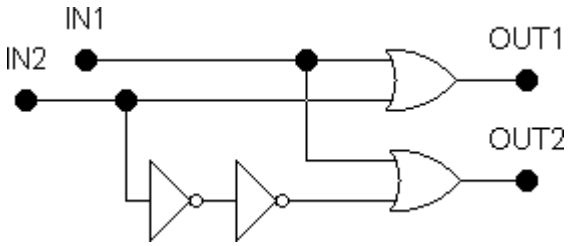
Элементтің оның кіріс сигналына байланысты әрекеті, яғни элементтің шығыс сигналының оның кіріс сигналының өзгерісіне байланысты өзгерісі, элементтің ішкі құралымымен (құрылым технологиясымен) анықталатын, нақтылы кідіріспен туады. Бұл құбылыс элементтің тезәрекеттілігін сипаттайды. Айтылған мәселені суреттеу мақсатында 1.1 суретте тікбұрышты сигналдың NOT элементі арқылы өту барысының уақыттық диаграммасы келтірілген.



1.1 сурет

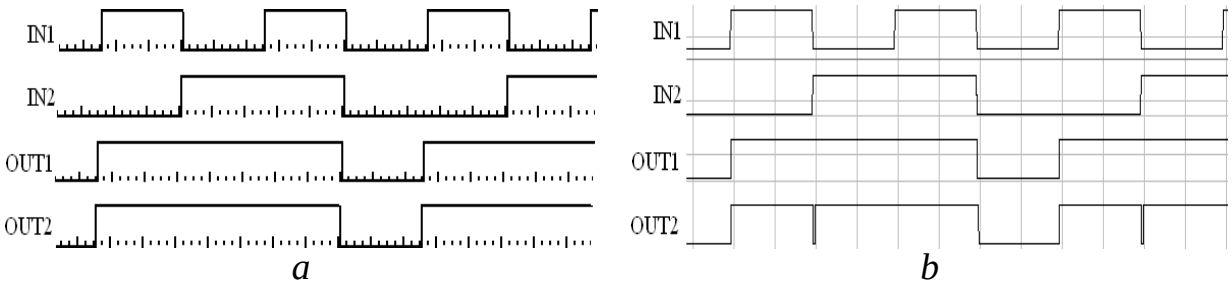
Суретте шығыс сигналының өзгеріс мезетінің кіріс сигналының өзгеріс мезетіне байланысты нақтылы уақытқа кідіретіндігі айқын көрініп тұр. Біздің Electronics Workbench модельдеу жүйесінде жүргізген өлшеміміз бойынша бұл кідірістің ұзақтығы 10 ns шамасында болады да, ол ТТЛ технологиясымен құрылған элементтердің осындай параметріне сәйкес келеді. Статикалық (яғни белгілі уақыт аралығында тиянақты мәнін сақтайтын) сигналдармен істейтін құрылғылардың жұмысына бұндай кідірістің байқарлықтай әсері болмайды. Бірақ кейбір жағдайларда (мысалы, тізбектеме құрылғыларда) бұндай кідірістің құрылғы жұмысына байқарлықтай әсер етуі мүмкін, демек оны ескеру керек болады.

Осындай кідірістің әрекетін суреттеу мақсатында екі OR элементінің кірістеріне қатар берілген екі сигналдың осы элементтер арқылы алынған логикалық қосылымын қарастыралық (1.2 суретті қара). Сұлбада көрсетілгендей, сигналдардың біреуі екінші элемент кірісіне екі NOT элементі арқылы берілген.



1.2 сурет

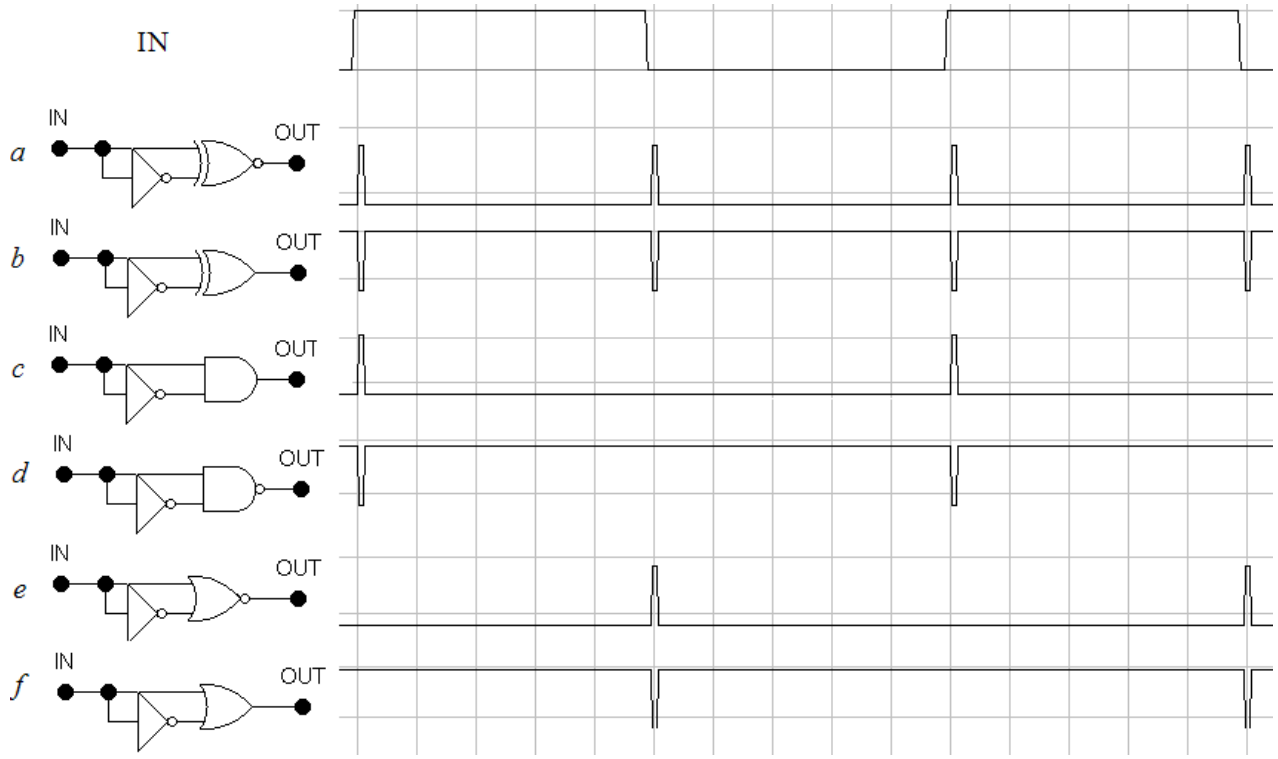
Идеалды жағдайда (яғни ЕМЕС элементтерінде ешқандай кідірістің болмауы кезінде) екі элементтің шығыстарындағы сигналдар (OUT1 және OUT2) бірдей болар еді (1.3, *a* сурет). Бұл диаграмма статикалық сигналдарды бақылауға арналған Electronics Workbench модельдеу жүйесіндегі, нақтылы инерциясы бар, Logic Analyzer аталымды арнайы аспап арқылы алынған.



1.3 сурет

Сезімтал осциллограф арқылы алынған диаграммада (1.3, *b* сурет) сигналдың екі ЕМЕС элементтері өтуге кеткен ~20 ns кідірісінің әсерінен осы элементтің шығысындағы сигнал құрамына бөгде импульстердің туғанын көрінеді. Шығыс сигналында құрамында осындай бөгде импульстердің тууы осы элемент шығысына қосылатын құрылғының бағдарланған жұмысын бүлдіруі мүмкін, сондықтан бұндай жағдайдың болмауын қамтамасыз ету керек болады.

Элементтердегі сигнал кідірісін іс жүзінде ұтымды пайдалануға да болады. Мысалы, түймежинамның жеке түймесін басылуы (оның жағдайының өзгертілуі) арқылы тудырылатын кіріс сигналының әрбір ауытқымасында (тура ауытқымасында да, теріс ауытқымасында да) нақтылы бағытты қысқа импульстер (ұзақтығы ~10 ns шамасындағы) қалыптастыру үшін NOR және XNOR элементтерінің негізіндегі қарапайым сұлбаларды пайдалануға болады (1.4, *a* және *b* суреттер).



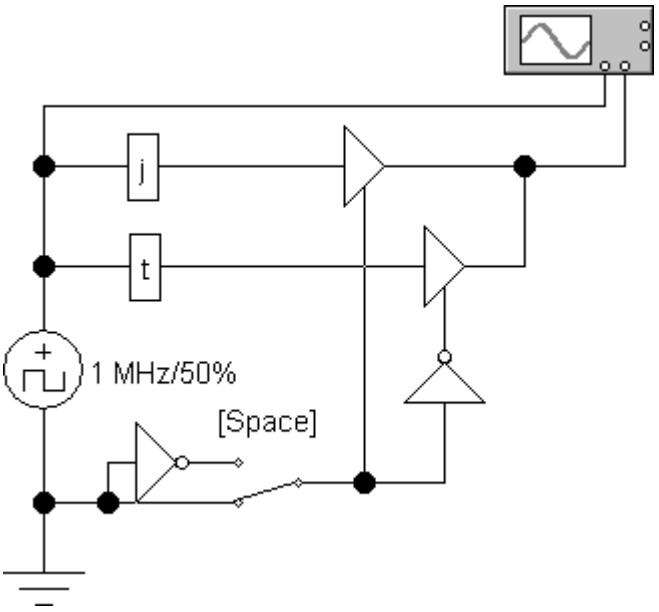
1.4 сурет

Өзге элементтер негізіндегі осы сияқты сұлбалар арқылы кіріс сигналының тура ауытқымасында ғана нақтылы бағытты қысқа импульстер қалыптастыруға болады (1.4, *c* және *d* суреттер). Осы сияқты импульстерді кіріс сигналының теріс ауытқымасында да қалыптастыруға болады (1.4, *e* және *f* суреттер)

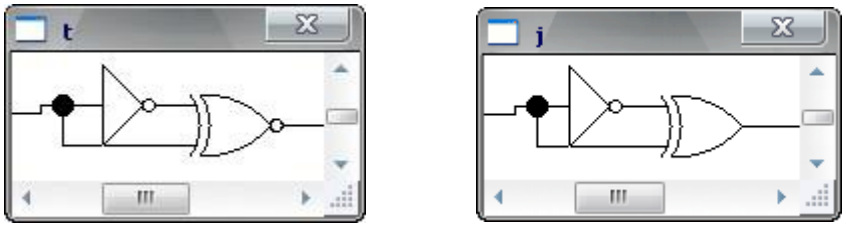
1.4 суретте келтірілген, қарапайым элементтер негізінде құрылған, сұлбалар іс жүзінде қолданымға аса ыңғайлы келуі мүмкін. Оған алда суреттелетін нақтылы құрылғылардың оқу мысалдары куә болады.

Әртүрлі қызметтік құрылғыларды ортақ желіге қосуға кеңінен қолданылатын, үшжағдайлы элементтердің мүмкіндіктерін де айта кету керек. 1.5 суретте t және j блоктары арқылы өткен сигналдарды сұлбаның ортақ шығысына (біздің жағдайда – осциллографқа) жіберуге мүмкіндік беретін сұлба келтірілген.

t және j блоктары ретінде, мысалы, XNOR және XOR элементтерінің негізінде құрылған және қосалқы сұлбалар (жабық блоктар) түрінде безендірілген қысқа импульс қалыптастырғыштарын пайдалануға болады (1.6 суретті қара). Бұл жағдайда Space ауыстырғышының жағдайын өзгерту арқылы сұлба шығысына қалыптастырылатын импульстердің үшжағдайлы элементтердің басқару кірісі бірлік жағдайдағысынан жіберілетініне көз жеткізуге болады.



1.5 сурет



1.6 сурет

1.2 Қиыстырма құрылғылар

Қиыстырма құрылғының сұлбасын құру, келесі әрекеттерді тізбелеп орындау арқылы, кезеңмен жүзеге асырылады:

– құрылатын құрылғының сөздік түсініктемесінің негізінде оның сәйкесті ақиқаттық кестесі құрылады;

– кесте түріндегі деректер негізінде сәйкесті логикалық өрнек жазылады;

– қажетті (және мүмкіндік болған) жағдайда алынған логикалық өрнек қарапайым түрге келтіріледі;

– алынған қарапайымдалған өрнек құруға жоспарланған түпнегіздік элементтер жинамына (core set) сәйкесті өрнекке түрлендіріледі;

– ақырғы логикалық өрнек бойынша құрылғы сұлбасы құрылады.

Мысал ретінде жұмысы 1.2 кесте арқылы берілген құрылғының құрылым тәртібін суреттелік.

1.2 к е с т е

$X2$	$X1$	$X0$	Y
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1

Кесте түрінде берілген функцияның логикалық өрнегін жазудың екі түрлі жолы (тәсілі) бар: көбейтінділердің қосылымы түрінде және қосындылардың көбейтілімі түрінде. Логикалық өрнектің сәйкесті жазылым түрлері дизъюнктивті және конъюнктивті түрдегі көрсетілімдер деп аталады.

Логикалық өрнектің дизъюнктивті түрдегі көрсетілімі келесі тәртіппен жүзеге асырылады:

– функцияның (y) бірлік мән қабылдайтын аргумент жинамдарына сәйкесті конъюнкциялар, яғни аргументтердің логикалық көбейтілімдері, жазылады;

– сәйкесті конъюнкциялардың (олар конъюнктивті термдер деп аталады), жазылымы кезінде бірлік мәндегі аргументтер тура түрінде, ал нөлдік мәндегі аргументтер теріс түрінде алынады;

– жазылған конъюнктивті термдер дизъюнкция арқылы біріктіріледі.

Сонымен, кесте түрінде берілген функцияға сәйкесті дизъюнктивті түрдегі логикалық өрнек аламыз:

$$\square = \overline{\square 2}\,\overline{\square 1}\square 0 \vee \overline{\square 2}\square 1\square 0 \vee \square 2\,\overline{\square 1}\square 0 \vee \square 2\square 1\square 0. \qquad (1.1)$$

Логикалық өрнектің конъюнктивті түрдегі көрсетілімі келесі тәртіппен жүзеге асырылады:

– функцияның (y) нөллік мән қабылдайтын аргумент жинамдарына сәйкесті дизъюнкциялар, яғни аргументтердің логикалық қосылымдары, жазылады;

– сәйкесті дизъюнкциялардың (олар дизъюнктивті термдер деп аталады), жазылымы кезінде нөллік мәндегі аргументтер тура түрінде, ал бірдік мәндегі аргументтер теріс түрінде алынады;

– жазылған дизъюнктивті термдер конъюнкция арқылы біріктіріледі.

Сонымен, кесте түрінде берілген функцияға сәйкесті конъюнктивті түрдегі логикалық өрнек аламыз:

$$\square = (\square 2 \vee \square 1 \vee \square 0)(\square 2 \vee \overline{\square 1} \vee \square 0)(\overline{\square 2} \vee \overline{\square 1} \vee \square 0). \qquad (1.2)$$

Құрылғының жұмысын сипаттаушы логикалық өрнектердің кез келгені арқылы оның сұлбасын құруға болады. Бірақ функция жазылымы күрделі болған сайын, оның сұлбасы да күрделі болатындығын ескеру керек. Демек

оларды мүмкіндігінше қарапайым түрге келтіруге тырысу керек болады. Логикалық функцияларды қарапайым түрге келтіруді (минимизациялауды) тікелей түрлендіру тәсілімен немесе Карно картасы арқылы жүзеге асыруға болады.

Логикалық функцияларды тікелей түрлендіру функция құрамындағы көрші термдерді тауып, оларға логика алгебрасының заңдары мен заңдылықтарын тікелей пайдаланып біріктіру арқылы жүргізіледі. Көрші термдерге бір аргументінің ғана айырмашылығы бар термдер жатады.

Осы тәсілдің жүзеге асырылымын (1.1) және (1.2) өрнектерінің түрлендірілуін қарастыру арқылы түсінуге болады.

(1.1) өрнегінің түрлендірілімі:

$$F = (\bar{x}_2\bar{x}_1 \vee \bar{x}_2 x_1 \vee x_2\bar{x}_1 \vee x_2 x_1) =$$

$$= \bar{x}_2(\bar{x}_1 \vee x_1) \vee x_2(\bar{x}_1 \vee x_1) =$$

$$= \bar{x}_2 \vee x_2 = 1.$$

(1.2) өрнегінің түрлендірілімі:

$$F = ((x_2 \vee x_1 \vee x_0)(x_2 \vee \bar{x}_1 \vee x_0))((x_2 \vee \bar{x}_1 \vee x_0)(\bar{x}_2 \vee \bar{x}_1 \vee x_0)) =$$

$$= ((x_2 \vee x_0)(x_1 \bar{x}_1))((\bar{x}_1 \vee x_0)(x_2 \bar{x}_2)) =$$

$$= (x_2 \vee x_0)(\bar{x}_1 \vee x_0) = x_2\bar{x}_1 \vee x_0.$$

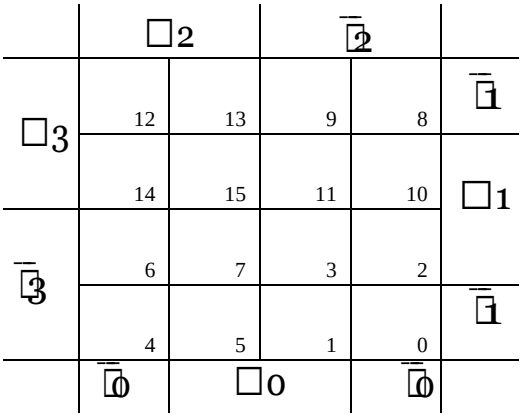
Логикалық функцияны Карно картасы арқылы қарапайымды түрге келтіру (минимизациялау) жолын суреттелік. Карно картасы – логикалық өрнектерді минимизациялауға ыңғайландырылған, функцияның кестелі суреттелімінің ерекше түрі. Қарастырылушы функция үшін құрылған Карно картасы 1.6-суретте келтірілген.

	x_1		$\bar{x}_1$	
x_2	0 (6)	1 (7)	1 (5)	1 (4)
$\bar{x}_2$	0 (2)	1 (3)	1 (1)	0 (0)
	$\bar{x}_0$	x_0		$\bar{x}_0$

1.6 сурет

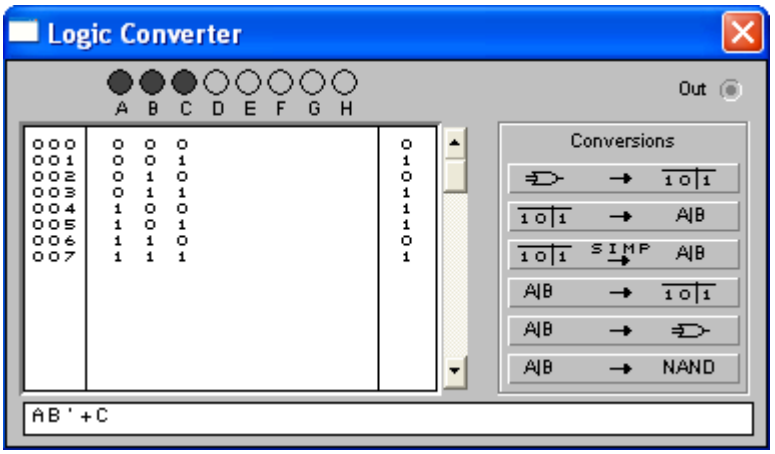
Карно картасында көрші термдер бірден көзге түседі: келтірілген картаның 1-, 3-, 5-, 7-ұяшықтарындағы бірліктердің өзара бірігіп, нәтижесінде одан $X0$ ғана қалатындығы және 4-ұяшықтағы бірліктің тек қана 5-ұяшықтағы бірлікпен бірігетіндігі көрініп тұр.

Төрт аргументті функцияға арналған Карно картасының түрі 1.7-суретте келтірілген.



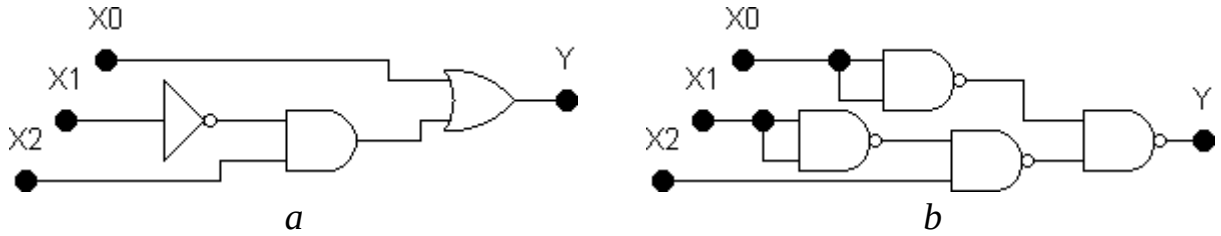
1.7 сурет

Electronics Workbench бағдарламасының құрамында кесте түрінде берілген функция өрнегінің толық және минимизацияланған түрлерін шығара алатын арнайы түрлендіргіш (Logic Converter) орналастырылған (1.8 сурет).



1.8 сурет

1.2 кестесі арқылы берілген құрылғының NOT, AND, OR элементтерінің негізінде құрылған сұлбасы 1.9, *а* суретте келтірілген.



1.9 сурет

Іс жүзінде құрылғы сұлбасын біртекті элементтер (мысалы, NAND элементтерінің) негізінде құру қажет болуы мүмкін. Бұндай жағдайда логикалық өрнекке де Морган заңы сәйкесті түрлендірілім (4-кезең) жүргізіледі:

$$\square = \square 2\,\overline{\square 1}\square 0 = \overline{\overline{\square 2}\square 0}$$

Логикалық түрлендіргіш арқылы алынған осы өрнекке сәйкесті құрылғы сұлбасы 1.9, *b* суретте келтірілген.

1.3 Қалыпты қиыстырма құрылғылар

Күрделі цифрлық құрылғылар, әдетте, қалыпты қызмет атқарушы жеке түрде құрылған қалыпты қызмет буындары арқылы құрылады. Осы тарауда қиыстырма құрылғылар түріндегі қалыпты қызмет буындарының құрылым принциптері мен жұмыс тәртібі қарастырылады.

1.3.1 Шифраторлар.

Шифратор (Coder) – түскен сигналға (дәлірек айтқанда, сигнал түскен кіріске) сәйкесті код қалыптастырушы құрылғы, яғни ол унитарлық кодты позициялық кодқа түрлендіреді. Мысал ретінде сегіз кірісті (X7 … X0) шифратордың сұлбасын құру жолын қарастыралық. Кіріс саны сегіз болғандықтан, ол үшразрядты код (*C*2 … *C*0) пен және осы кодты жады буферіне жазуға мүмкіндік беретін құптау сигналын (*O*) қалыптастыру керек.

Үлкен құрылымдардың құрамындағы жеке қызмет буындары, әдетте, кезектеп іске қосылады, бұл олардың іске қосу кірісіне сәйкесті деңгейлі сигнал жіберілуі арқылы жүзеге асырылады. Осындай іске қосу кірісі (I, Initialization) біздің құрастыратын шифраторда да ескерілгені дұрыс. Тағы бір ескеретін мәселе: қалыпты қызмет буындарының іске қосу кірісі мен құптау шығысындағы сигналдың жандандыру деңгейі нөлдік мәнінде алынады. Шифратордың ақпараттық кірістеріне түсетін сигналдардың да жандандыру деңгейі нөлдік мәнінде болғаны бұндай құрылғыны іс жүзінде құруға ыңғайлы болады.

Шифратордың келтірілген түсіндірме суреттемесі оның ақиқаттық кестесін құруға толық мәлімет береді, келтірілген түсіндірме мәліметтерінің негізінде сол кестені құрамыз (1.3 кесте).

1.3 к е с т е

I	X_i	C_2	C_1	C_0	O
0	0	0	0	0	0
	1	0	0	1	0
	2	0	1	0	0
	3	0	1	1	0
	4	1	0	0	0
	5	1	0	1	0
	6	1	1	0	0
	7	1	1	1	0
1	*	0	0	0	1

Кесте деректерінің негізінде жазылған шифратор шығыстарының логикалық өрнектері:

$$C0 = \overline{(\overline{X7} \vee \overline{X5} \vee \overline{X3} \vee \overline{X1})} = X7 \vee X5 \vee X3 \vee X1$$

$$C1 = \overline{(\overline{X7} \vee \overline{X6} \vee \overline{X3} \vee \overline{X2})} = X7 \vee X6 \vee X3 \vee X2$$

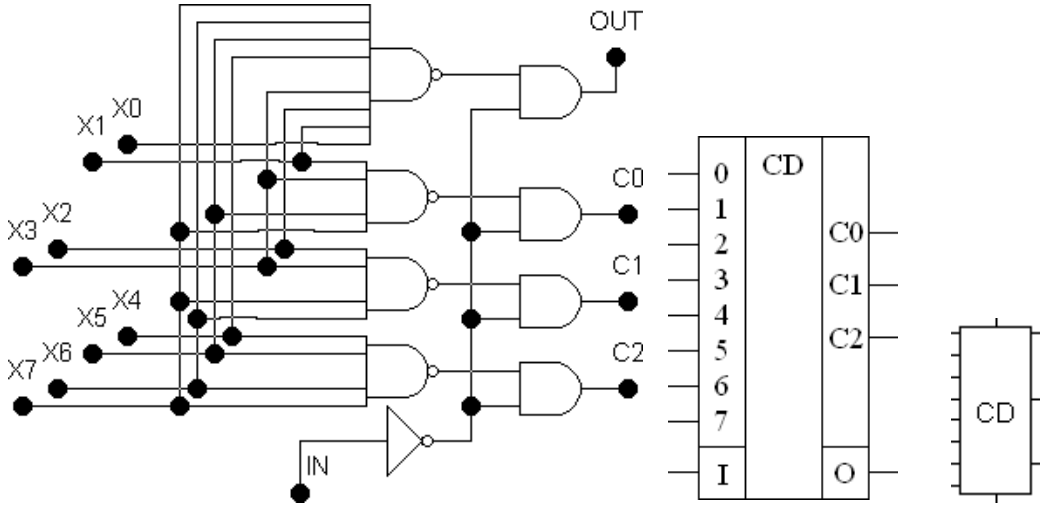
$$C2 = \overline{(\overline{X7} \vee \overline{X6} \vee \overline{X5} \vee \overline{X4})} = X7 \vee X6 \vee X5 \vee X4$$

$$OUT = \overline{(\overline{X7} \vee \overline{X6} \vee \overline{X5} \vee \overline{X4} \vee \overline{X3} \vee \overline{X2} \vee \overline{X1} \vee \overline{X0})} =$$

$$= X7 \vee X6 \vee X5 \vee X4 \vee X3 \vee X2 \vee X1 \vee X0$$

Бұл өрнектер, алдымен, дизъюнктивті түрде жазылып, сосын де Морган заңын пайдалану арқылы сұлбаны ЖӘНЕ-ЕМЕС элементтері арқылы құруға арналған түріне түрлендірілген. Сұлбаларды ЖӘНЕ-ЕМЕС элементтері құруға басты себеп ретінде ЖӘНЕ-ЕМЕС элементтерінің ішкі құрылымына байланысты олардың тезәрекеттілігінің басқа элементтермен салыстырғанда жоғары болуынан екендігін айта кету керек. 1.10, *a*-суретте шифратордың алынған өрнектер негізінде құрылған сұлбасы келтірілген.

Шифратор цифрлық жүйелердің сұлбаларында шартты сызба белгілемесімен (1.10, *b*-сурет) көрсетіледі. 1.10, *c*-суретте Electronics Workbench бағдарламасының мүмкіндігін пайдалану арқылы жүзеге асырылған алдыңғы жиналған сұлбаның біріктірілген жеке блок (Subcircuit) түріндегі суреттемесі келтірілген (оның сәйкесті шықпалары олардың келтірілген құрылым сұлбасындағы орналастырылым бағытына сай шығарылған).



a *b* *c*

1.10 сурет

Жоғарыла құрылған шифратор сұлбасының айта кететін бір кемшілігі бар, оған екі сигнал қатар жіберілген жағдайда екі сигнал коды араласып, нәтижесінде түскен сигналдардың ешқайсысына да сәйкес келмейтін код

қалыптасады. Осындай жағдайды болдырмас үшін шифратордың өндірісте шығарылатын микросұлбалары (мысалы, 74148 микросұлбасы), әдетте, мәртебелі шифраторлар түрінде құрылады. Яғни олар түскен бірнеше сигналдың белгіленген мәртебесі жоғарғысының кодын шығарады да, қалғандарына көңіл бөлмейді.

1.3.2 Дешифраторлар.

Дешифратор (Decoder) – кірістеріне түскен екілік кодқа сәйкесті шығысында сигнал қалыптастырушы құрылғы, яғни ол позициялық кодты унитарлық кодқа түрлендіреді.. Мысал ретінде төрт теріс шығысында ($Q3 \ldots Q0$) теріс сигналдар қалыптастыратын дешифратордың сұлбасын құру жолын қарастыралық. Шығыс саны төртеу болғандықтан, оның кірісіне түсетін код екіразрядты ($A1A0$) болады. Дешифратордың іске қосу кірісіндегі (E, Enable) сигналдың жандандыру деңгейін нөлдік мәнінде алалық.

Дешифратордың келтірілген түсіндірме суреттемесінің негізінде оның ақиқаттық кестесін (1.4 кесте) құрамыз.

1.4 к е с т е

E	*A1*	*A0*	*Q3*	*Q2*	*Q1*	*Q0*
0	0	0	1	1	1	0
	0	1	1	1	0	1
	1	0	1	0	1	1
	1	1	0	1	1	1
1	*	*	1	1	1	1

Кесте деректерінің негізінде жазылған дешифратор шығыстарының логикалық өрнектері:

$$\square o = \overline{\square\square\square}\,\square\phi$$

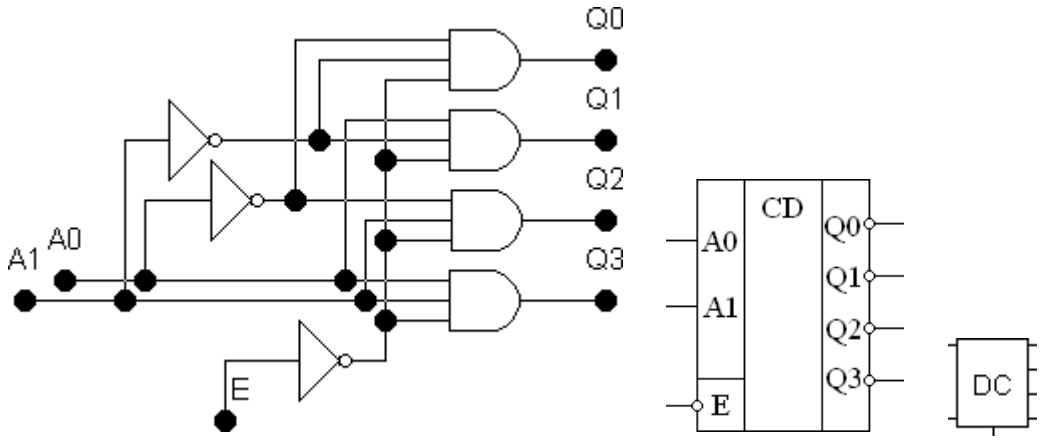
$$\square o = \overline{\square\square}\,\square o$$

$$\square o = \overline{\square\square}\,\mathbf{1}\,\overline{\square}\phi$$

$$\square o = \overline{\square\square}\,\mathbf{1}\,\square o.$$

1.11, *a*-суретте дешифратордың жазылған өрнектер негізінде құрылған сұлбасы келтірілген.

Дешифраторлар сұлбада шартты сызба белгілемесімен (1.11, *b*-сурет) көрсетіледі. 1.11, *c*-суретте дешифратордың Electronics Workbench бағдарламасында жиналған сұлбасының біріктірілген жеке блок түріндегі суреттемесі келтірілген.

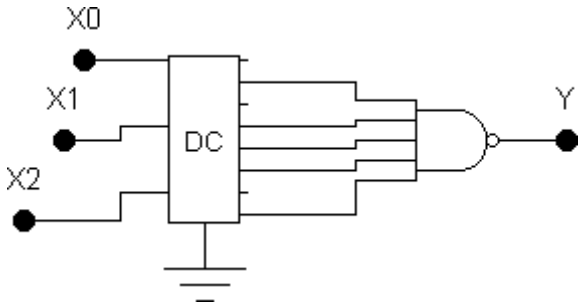


a *b* *c*

1.11 сурет

1.2 тарауда қарастырылған қиыстырма құрылғының шығыс сигналын суреттеуші логикалық функцияның өрнегіне (1.1) көз салсақ, оның құрамына кіретін конъюнктивті термдердің сегіз тура шығысты дешифратордың сәйкесті шығыстарының логикалық өрнектеріне сәйкес келетіндігін, ал функция өрнегінің өзінің, шындығында, сәйкесті дешифратордың жұмыс логикасын анықтайтынын байқауға болады. Демек, дешифратордың сәйкесті шығыстарын OR элементі арқылы біріктіріп, осы құрылғыны жүзеге асыруға болады. Теріс шығысты дешифратор пайдаланылуы жағдайында (біздің жағдайдағыдай) (1.1) өрнегін де Морган заңы бойынша түрлендіру керек болады:

Түрлендіру нәтижесінде алынған өрнектен берілген құрылғының қызметін жүзеге асыру үшін дешифратордың сәйкесті шығыстарын біріктіруге NAND элементін пайдалану қажеттігіне көз жеткіземіз (1.12 суретті қара).



1.12 сурет

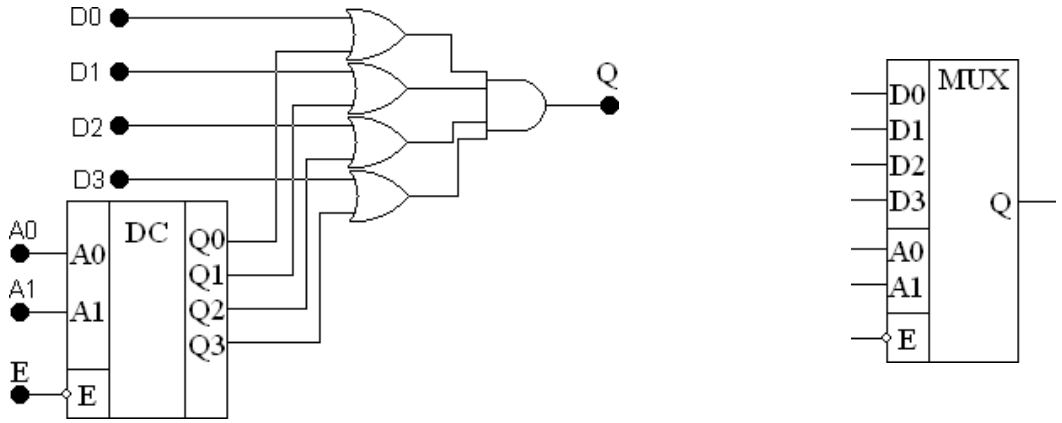
Қиыстырма құрылғыларды дешифратор негізінде құру – аса ыңғайлы тәсіл: біріншіден, логикалық өрнекті қарапайым түрге келтірудің қажеті жоқ

(дәлірек айтқанда, қажетті жалғанымдарды логикалық өрнектің өзінің де қажеті жоқ – қажетті жалғанымдарды кестеден-ақ анықтауға болады), екіншіден, жалғыз дешифратор негізінде функцияны қатар жүзеге асыруға болады.

1.3.3 Мультиплексорлар.

Мультиплексор (Multiplexer) – ақпараттық сигналдардың түскен адрестік кодқа сәйкесті біреуін шығысына қосылған құрылғыға бағыттаушы құрылғы. Бұл анықтамаға сәйкесті мультиплексордың екі кірістер тобы ажыратылады, олар: ақпараттық және адрестік кірістері; әрине, оның іске қосылу кірісі де болу керек.

Мультиплексордың дешифратор арқылы жүзеге асырылған сұлбасы мен шартты сызба белгілемесі 1.13 суретте келтірілген.



a *b*

1.13 сурет

Бірнеше мультиплексорды қатар қосу арқылы бірнешеразрядты (мысалы, сегізразрядты) сөздердің біреуін бір арнаға жіберу жұмысын атқаратын мультиплексорлық құрылым құру қиын емес, бұндай құрылым арналы мультиплексор деп аталады.

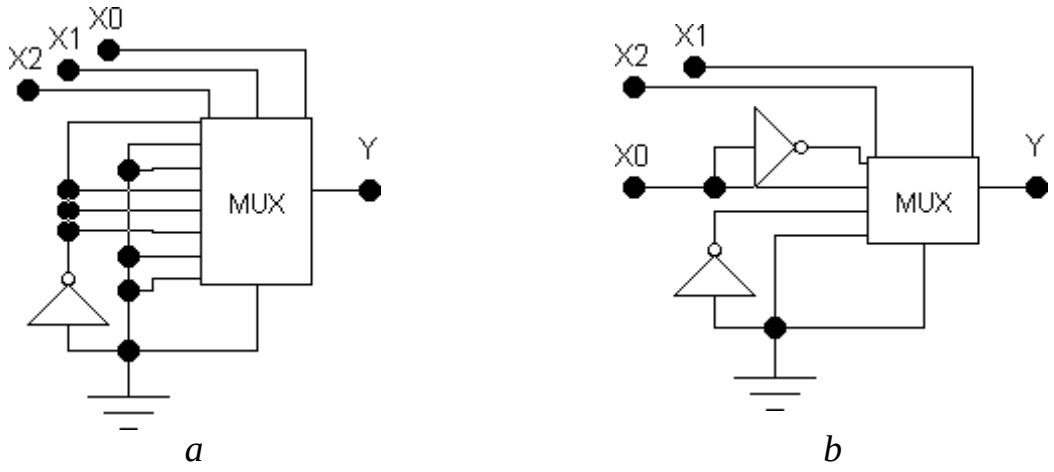
1.5 к е с т е

X2	X1	X0	Y
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	0

Мультиплексор негізінде де қиыстырма құрылғының жұмысын жүзеге асыруға болады. Мысал ретінде 1.5 кестеде келтірілген функцияның

мультиплексор негізінде жүзеге асырылымының екі жолын (тәсілін) қарастырамыз.

Бірінші тәсіл мультиплексордың адрестік кірістерінің саны жүзеге асырылатын функцияның аргументтерінің санымен бірдей болған жағдайда қолданылады. Бұл жағдайда мультиплексордың ақпараттық кірістері функцияның сәйкесті аргументтер жинағындағы мәндеріне сәйкесті қойылады да, оның адрестік кірістеріне аргумент мәндеріне сәйкесті сигналдар беріледі. Берілген функцияның осы тәсілмен сегіз кірісті мультиплексор негізінде құрылуы 1.14, *a* суретте көрсетілген. Бұнда пайдаланылған сәйкесті қосалқы сұлбаның ақпараттық кірістері – оның сол жағына, ал адрестік кірістері оң жағына орналастырылған.



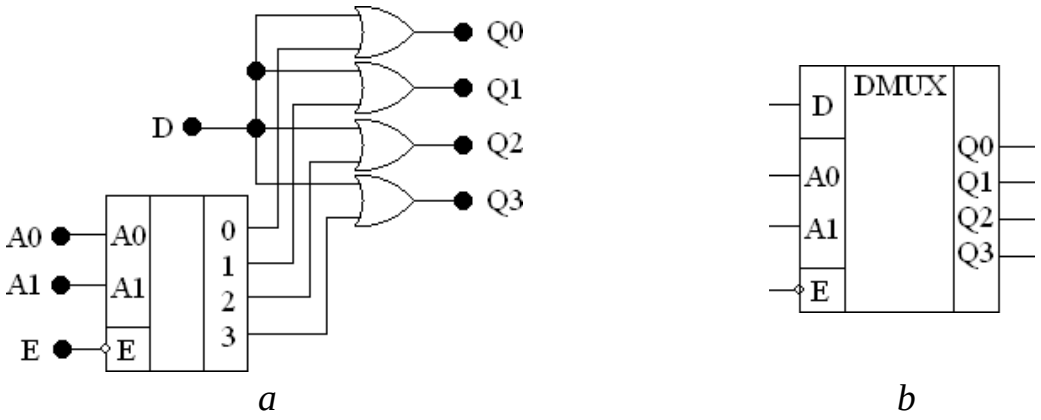
a *b*

1.14 сурет

Екінші тәсіл функцияны адрестік кірістерінің саны аргументтер санынан кем болатын мультиплексор арқылы құру қажеттігінде қолданылады. Бұл жағдайда мультиплексордың адрестік кірістеріне аргументтердің бір бөлігіне сәйкесті сигналдар беріледі де, ал қалғаны мультиплексордың ақпараттық кірістерін баптауға пайдаланылады. Келтірілген функцияның осы тәсілмен төрт кірісті мультиплексор негізінде құрылуы 1.14, *b* суретте көрсетілген.

1.3.4 Демультиплексорлар.

Демультиплексор (Demultiplexer) – дерек кірісіндегі сигналды сілтеу коды арқылы анықталған шығысына жіберетін құрылғы. Демультиплексордың дешифратор арқылы құрылған сұлбасы мен оның сәйкесті шартты сызба белгілемесі 1.15 суретте келтірілген.



a *b*

1.15 сурет

1.3.5 Қосуыштар.

Қосуыш – екі санға арифметикалық қосу операциясының жүзеге асырылуын орындаушы құрылғы. Көпразрядты сандарды қосу бірнеше бірразрядты қосуыштар арқылы жүзеге асырылады. Сондықтан, алдымен, сол бірразрядты қосуыштартардың құрылым принциптерін қарастырамыз.

Екі бірразрядты санды қосу кезінде олардың қосынды нәтижесі (S, Sum) мен келесі разрядқа жіберілетін тасымал (C, Carry) шығарылу керек, демек оның жұмысын 1.6 кестемен суреттеуге болады.

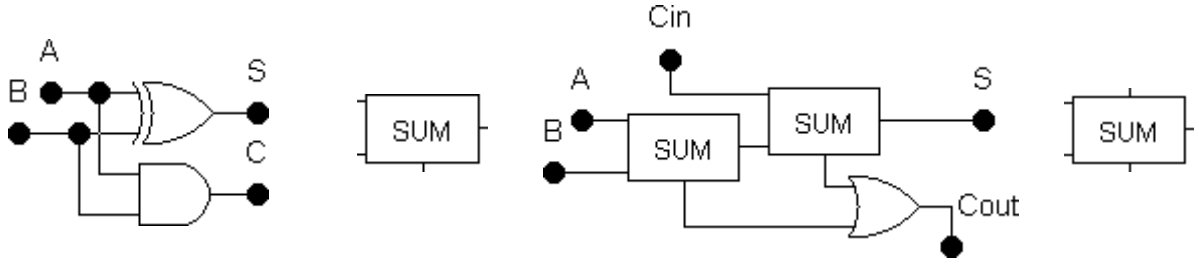
1.6 к е с т е

A	B	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Кесте деректері негізінде жазылған құрылғы шығыстарының логикалық өрнектері:

$$S = A \oplus B; \quad C = AB.$$

Осы өрнектердің негізінде құрылған қосуыштың сұлбасы мен оның сәйкесті біріктірмесі 1.16, *а* суретте келтірілген.

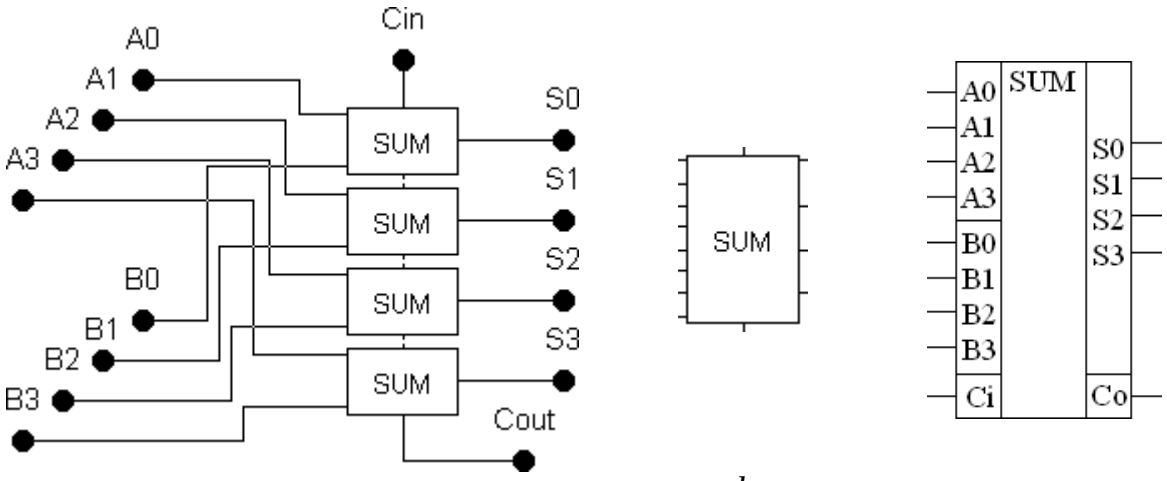


a *b*

1.16 сурет

Бұл қосуыш көпразрядты сандардың бірінші разрядтарын қосуға ғана жарайды (сондықтан ол жартылай қосуыш деп аталады), ал оның қалған разрядтарға жарауы үшін оның алдыңғы разрядтан түскен тасымалды да қосатын мүмкіндігі болу керек. Толық қосуыштың жартылай қосуыштар арқылы құрылған сұлбасы мен оның сәйкесті біріктірмесі 3.10, *b*-суретте келтірілген.

Көпразрядты қосуыштарды суреттелген толық қосуыштар арқылы құру қиын емес. 1.17 суретте төртразрядты қосуыштың сұлбасы, біріктірме блогы және шартты сызба белгілемесі келтірілген.



a *b* *c*

1.17 сурет

1.4 Тізбектеме құрылғылар

Тізбектеме құрылғылардың жоғарыда келтірілген анықтамасында көрсетілгендей, олардың қатарына жадылы құрылғылар, яғни алдыңғы жағдайын сақтау мүмкіндігі бар құрылғылар жатады. Сондықтан, алдымен, жады элементтерінің (триггерлердің) түрлерімен және олардың құрылымы мен жұмыс принциптерімен танысамыз.

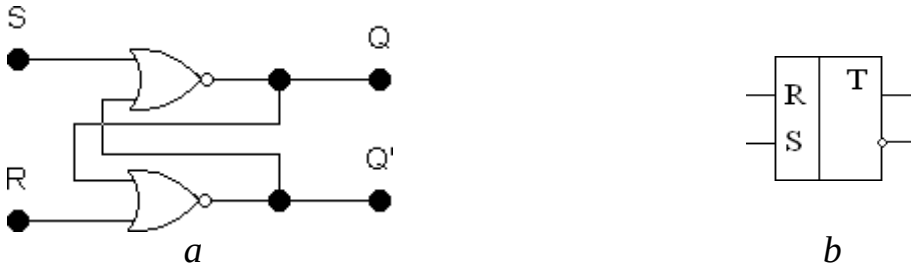
1.4.1 Жады элементтері – триггерлер.

Триггерлер – екі тиянақты жағдайы бар бірразрядты екілік сандардың мәнін сақтауға арналған құрылғылар. Оның тиянақты жағдайларының біреуі лог.1-ге, екіншісі лог.0-ге сәйкес келеді. Әдетте, триггерлік құрылымның қарама-қарсы деңгейлі (тура және теріс деп аталатын) екі шығысы болады, триггердің жағдайы оның тура шығысындағы сигнал деңгейімен анықталады.

Ақпараттың енгізілу (жазылу) тәртібіне байланысты триггерлер асинхронды және синхронды болып бөлінеді.

Асинхронды триггерлердің жағдайының өзгеруі олардың кірісіне берілетін ақпараттық сигналдардың өзгерісімен ғана анықталады. Оларды NOR немесе NAND элементтерінің негізінде құруға болады.

Асинхронды триггердің NOR элементтерінің негізінде құрылған сұлбасы 1.18, *а* суретте келтірілген.



а *b*

1.18 сурет

Бұл құрылымның жұмысын 1.7 кестедегі жазылым ретімен талқылау арқылы қарастырамыз.

1.7 к е с т е

	R	S	Q	Q’
1	0	0	?	?
2	1		0	1
3	0		1	1
4		1	1	0
5		0	1	0
6	1	1	0	0
7	0	0	???	???

Кестенің бірінші жолында – R = 0, S = 0; екі элементтің де бір кірісінде 0 тұр, ал олардың кері байланыстық кірістерінде қарастырылым басында қандай сигнал тұрғаны белгісіз, демек құрылым шығыстарындағы (Q және Q’) сигнал деңгейлері де белгісіз.

Кестенің екінші жолында – R = 1, S = 0; жоғарғы элементтің бір кірісінде лог.1 тұрғандықтан, бұл элементтің жұмыс логикасына сәйкесті, оның шығысында (Q) лог.0 деңгейлі сигнал туады да, ол кері байланыс жолы арқылы төменгі элементтің екінші кірісіне түседі, нәтижесінде оның шығысында (Q’) лог.1 деңгейлі сигнал туады; сонымен, бұл жағдайда триггер лог.0 жағдайына келтіріледі (яғни оған 0 жазылады).

Кестенің үшінші жолында – R = 0, S = 0; жоғарғы элементтің R кірісіндегі сигнал деңгейі өзгергенмен, оның екінші кірісінде (құрылымның Q’ шығысына жалғанған) лог.1 деңгейі тұрғандықтан, бұл элементтің шығысындағы сигнал деңгейі алдыңғы лог.0 деңгейінде қалады, демек триггердің жағдайы өзгермейді.

Кестенің төртінші жолында – R = 0, S = 1; төменгі элементтің бір кірісінде лог.1 тұрғандықтан, оның шығысында (Q’) лог.0 деңгейлі сигнал туады да, ол кері байланыс жолы арқылы жоғарғы элементтің екінші кірісіне түседі, нәтижесінде оның шығысында (Q) лог.1 деңгейлі сигнал туады;

сонымен, бұл жағдайда триггер лог.1 жағдайына келтіріледі (яғни, оған 1 жазылады).

Кестенің бесінші жолында – R = 0, S = 0; төменгі элементтің S кірісіндегі сигнал деңгейі өзгергенмен, оның құрылымның Q шығысына жалғанған кірісінде лог.1 деңгейі тұрғандықтан, бұл элементтің шығысындағы сигнал деңгейі алдыңғы лог.1 деңгейінде қалады, демек триггердің алдыңғы жағдайы сақталады.

Құрылымның қарастырылған жағдайларынан келесі қорытынды шығаруға болады: R = 1, S = 0 берілу жағдайы триггердің 0 жазу режиміне, R = 0, S = 1 жағдайы триггердің 1 жазу режиміне, ал R = 0, S = 0 жағдайы триггердің жазылған санды сақтау режиміне сәйкес келеді.

Триггердің кірістеріне R = 1, S = 1 берілген кезде (алтыншы жол) екі элементтің де шығыстарында лог.0 деңгейлі сигнал туады да, триггер сақтау режиміне ауыстырылғанда (жетінші жол), триггердің қай жағдайға тиянақталатындығы – кездейсоқ оқиға. Демек бұндай жағдайды тудырмас үшін, қарастырылған триггердің кірістеріне R = 1, S = 1 берілмеуі керек, сондықтан бұл жағдай триггердің тыйым салынған режимі деп аталады.

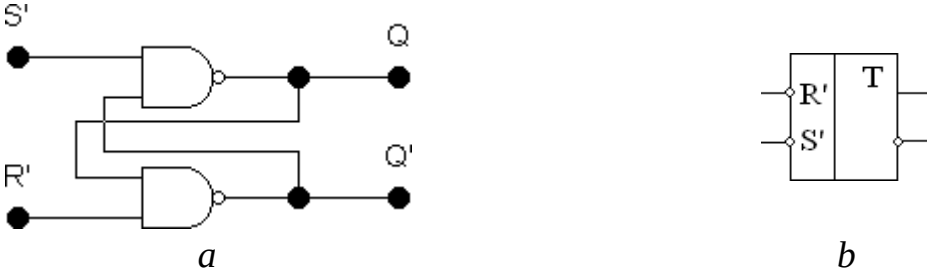
Сонымен, қарастырылған триггердің жұмыс режимдерін 1.8 кесте түрінде суреттеуге болады.

1.8 к е с т е

R	S	Q	Q’	Жұмыс режимдері
0	0	Q_0	Q_0’	Сақтау режимі
0	1	1	0	‘1’ жазу режимі
1	0	0	1	‘0’ жазу режимі
1	1	–	–	Тыйым салынған режим

Қарастырылған триггердің жазу режимдері сәйкесті ақпараттық сигналдардың (S – Set, R – Reset) бірлік деңгейімен жүзеге асырылатындығына байланысты бұндай триггер тура кірісті триггер деп аталады. Оның шартты сызба белгілемесі 1.18, суретте келтірілген.

NAND элементтерінің негізінде құрылған асинхронды триггердің сұлбасы 1.19, *а* суретте келтірілген.



a *b*

1.19 сурет

Бұл құрылымның жұмысын 1.9 кестедегі жазылым ретімен қарастырамыз.

1.9 к е с т е

	R	S	Q	Q’
1	1	1	?	?
2	0		0	1
3	1		0	1
4		0	1	0
5		1	1	0
6	0	0	1	1
7	1	1	???	???

Кестенің бірінші жолында – R = 1, S = 1; екі элементтің де бір кірісінде 1 тұр, ал олардың кері байланыстық кірістерінде қарастырылым басында қандай сигнал тұрғаны белгісіз, демек құрылым шығыстарындағы (Q және Q’) сигнал деңгейлері де белгісіз.

Кестенің екінші жолында – R = 0, S = 1; төменгі элементтің бір кірісінде лог.0 тұрғандықтан, бұл элементтің жұмыс логикасына сәйкесті, оның шығысында (Q’) лог.1 деңгейлі сигнал туады да, ол кері байланыс жолы арқылы жоғарғы элементтің екінші кірісіне түседі, нәтижесінде оның шығысында (Q) лог.0 деңгейлі сигнал туады; сонымен, бұл жағдайда триггер лог.0 жағдайына келтіріледі (яғни оған 0 жазылады).

Кестенің үшінші жолында – R = 1, S = 1; төменгі элементтің R кірісіндегі сигнал деңгейі өзгергенмен, оның құрылымның Q шығысына жалғанған кірісінде лог.0 деңгейі тұрғандықтан, бұл элементтің шығысындағы сигнал деңгейі алдыңғы лог.1 деңгейінде қалады, демек триггердің жағдайы өзгермейді.

Кестенің төртінші жолында – R = 1, S = 0; жоғарғы элементтің бір кірісінде лог.0 тұрғандықтан, оның шығысында (Q) лог.1 деңгейлі сигнал туады да, ол кері байланыс жолы арқылы төменгі элементтің екінші кірісіне түседі, нәтижесінде оның шығысында (Q’) лог.0 деңгейлі сигнал туады; сонымен, бұл жағдайда триггер лог.1 жағдайына келтіріледі (яғни, оған 1 жазылады).

Кестенің бесінші жолында – R = 1, S = 1; жоғарғы элементтің S кірісіндегі сигнал деңгейі өзгергенмен, оның құрылымның Q’ шығысына жалғанған кірісінде лог.0 деңгейі тұрғандықтан, бұл элементтің шығысындағы сигнал деңгейі алдыңғы лог.1 деңгейінде қалады, демек триггердің алдыңғы жағдайы сақталады.

Құрылымның қарастырылған жағдайларынан келесі қорытынды шығаруға болады: R = 0, S = 1 берілу жағдайы триггердің 0 жазу режиміне, R = 1, S = 0 жағдайы триггердің 1 жазу режиміне, ал R = 1, S = 1 жағдайы триггердің жазылған санды сақтау режиміне сәйкес келеді.

Триггердің кірістеріне R = 0, S = 0 берілген кезде (кестенің алтыншы жолы) екі элементтің де шығыстарында лог.1 деңгейлі сигнал туады да, триггер сақтау режиміне ауыстырылғанда (кестенің жетінші жолы), триггердің қай жағдайға тиянақталатындығы кездейсоқ оқиға. Демек бұндай

жағдайды тудырмас үшін, қарастырылған триггердің кірістеріне R = 0, S = 0 берілмеуі керек, сондықтан бұл жағдай триггердің тыйым салынған режимі болады. Сонымен, қарастырылған триггердің жұмыс режимдерін 1.10 кесте түрінде суреттеуге болады.

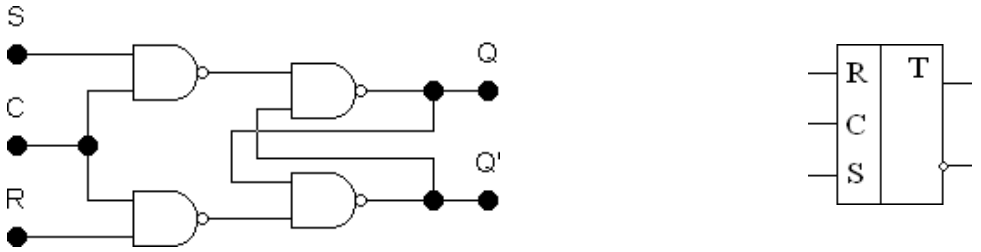
1.10 к е с т е

R	S	Q	Q’	Жұмыс режимдері
0	0	–	–	Тыйым салынған режим
0	1	1	0	‘0’ жазу режимі
1	0	0	1	‘1’ жазу режимі
1	1	Q_0	Q_0’	Сақтау режимі

Қарастырылған триггердің жазу режимдері сәйкесті ақпараттық сигналдардың (S – Set, R – Reset) нөлдік деңгейімен жүзеге асырылатындығына байланысты бұндай триггер теріс кірісті триггер деп аталады. Оның шартты сызба белгілемесі 1.19, *b* суретте келтірілген.

Енді синхронды триггерлерді қарастыралық. Синхронды триггерлердің жаңа жағдайға ауысуы арнайы басқару сигналымен жүзеге асырылады. Басқару сигналын қабылдау тәсіліне байланысты синхронды триггерлер статикалы басқарылымды және динамикалы басқарылымды түрлеріне ажыратылады.

Статикалы басқарылымды (яғни деңгей арқылы басқарылатын) RS-триггердің сұлбасын теріс кірісті асинхронды RS-триггерге екі NAND элементінен тұратын басқару сұлбасын қосу арқылы жүзеге асыруға болады (1.20, *a* сурет).



a *b*

1.20 сурет

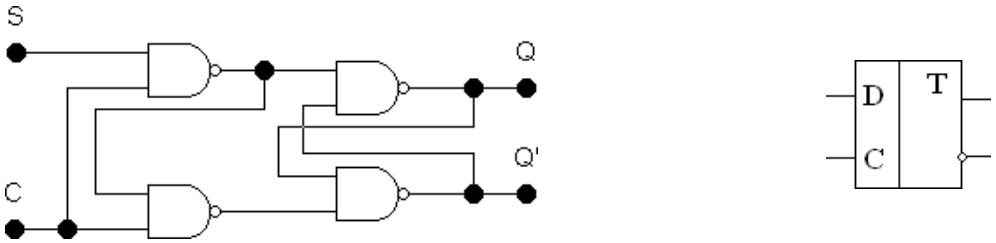
Бұл триггердің жұмыс режимдері 1.11 кестеде суреттелген.

1.11 к е с т е

C	R	S	Q	Q’	Жұмыс режимдері
0	x	x	Q_0	Q_0’	Сақтау режимі
1	0	0	–	–	Сақтау режимі
	0	1	1	0	‘1’ жазу режимі
	1	0	0	1	‘0’ жазу режимі
	1	1	Q_0	Q_0’	Тыйым салынған режим

C = 0 кезінде триггер сақтау режимінде болады. Триггердің R мен S кірістеріндегі сигнал деңгейлеріне сәйкесті жағдайға ауысуы C = 1 кезінде жүзеге асырылады. Синхронды RS-триггердің шартты сызба белгілемесі 1.21, *b* суретте келтірілген.

Статикалы басқарылымды D-триггердің (Delay – кідіріс) сұлбасы мен шартты сызба белгілемесі 1.21 суретте келтірілген, ал оның жұмыс режимдері 1.12 кестеде суреттелген.



a *b*

1.21 сурет

1.12 к е с т е

C	D	Q	Q’	Жұмыс режимдері
0	x	Q_0	Q_0’	Сақтау режимі
1	0	0	1	‘0’ жазу режимі
	1	1	0	‘1’ жазу режимі

Бұдан әрі динамикалы басқарылатын триггерлердің үш түрін қарастырамыз, олар – D-триггер, JK-триггер және T-триггер.

Динамикалы басқарылатын триггерлердің жағдайының ауыстырылуы басқару сигналының сәйкесті (тура немесе теріс) ауытқымасымен жүзеге асырылады. Динамикалы басқарылатын триггерлер, әдетте, үштриггерлі құрылым негізінде құрылады. Олардың сұлбақұралымына көңіл аудармай-ақ, жұмыс режимдерімен ғана танысамыз.

Динамикалы басқарылымды D-триггердің шартты сызба белгілемесі 1.22, *a* суретте келтірілген, ал оның жұмыс режимдері 1.13 кестеде суреттелген.

a *b*

1.22 сурет

1.13 к е с т е

C	D	Q	Q’	Жұмыс режимдері
⎍	0	0	1	Сақтау режимі
	1	1	0	Жазу режимі

Бұл триггердің жағдайының ақпараттық сигналдың мәніне (D) сәйкесті ауысуы басқару сигналының тура ауытқу (0-ден 1-ге) мезетінде жүзеге асырылады. Оның шартты сызба белгілемесінде басқару сигналының ауытқу бағыты сәйкесті бағыттама түрінде бейнеленген.

JK-триггердің төрт түрлі жұмыс режимі бар (1.14 кесте): оның үш режимі (‘0’ жазу, ‘1’ жазу, сақтау) RS-триггердегі сияқты, ал төртінші режим санау режимі деп аталады, бұл режимде триггер кезекті жағдайынан қарсы жағдайға ауысады. JK-триггердің келтірілген жұмыс кестесінде және оның шартты сызба белгілемесінде (1.22, *b* сурет) оның жағдайын ауыстыруы басқару сигналының теріс ауытқу (1-ден 0-ге) мезетінде жүзеге асырылатындығы сәйкесті бағыттау белгілерімен суреттелген.

1.14 к е с т е

C	J	K	Q	Q’	Жұмыс режимдері
⌐_	0	0	Q_0	Q_0’	Сақтау режимі
	0	1	0	1	‘0’ жазу режимі
	1	0	1	0	‘1’ жазу режимі
	1	1	Q_0’	Q_0	Санау режимі

Т-триггер – санау режимінде істейтін триггер. Т-триггердің қызметін жүзеге асыру үшін D-триггер мен JK-триггердің қосылым тәртібі 1.23 суретте келтірілген.

1.23 сурет

Микросұлба түрінде D-триггерлер мен JK-триггерлер шығарылады. Бұл микросұлбаларда ақпараттық және басқару сигналдарын енгізу шықпаларымен қатар, бастапқы қойылым сигналдарының (R, S) шықпалары орналастырылады (4.7 сурет). Бұл кірістерге берілетін асинхронды сигналдардың мәртебесі басқа сигналдарға қарағанда жоғары болады.

1.24 сурет

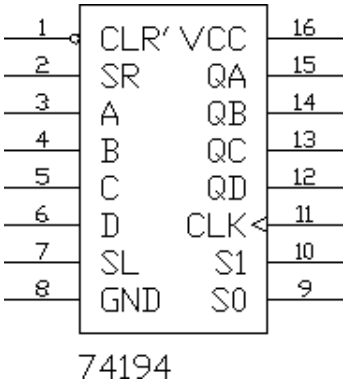
1.4.2 Регистрлер.

Регистр – көпразрядты екілік сан сақтауға арналған құрылғы. Олар триггерлердің (мысалы D-триггерлердің) негізінде құрылады. Әрине, регистр құрамындағы триггерлердің саны регистрдің разряд санымен анықталады. Регистрге санның енгізілу тәртібіне байланысты олардың тізбекті және параллель регистр деп аталатын түрлері болады.

Тізбекті регистрде әрбір триггердің шығысы келесі триггердің ақпараттық кірісіне жалғанады. Бірнеше разрядты екілік сан түріндегі сөздің разряд мәндері бір-бірлеп бірінші триггерге тактілік сигналдың сәйкесті ауытқуында енгізіледі де, осы мезетте триггерлердегі бұрынғы енгізілген разряд мәндері бір триггерден келесі триггерге ығысып отырады. Әрине, бүкіл сөзді енгізу үшін, оның қанша разряды болса, сонша тактілік сигнал қажет болады.

Параллель регистрдің триггерлерінің ақпараттық кірістеріне сөз разрядтарының мәндері қатар беріліп, олар регистрге жалғыз тактілік сигнал арқылы енгізіледі.

Екі түрлі енгізу тәсілі қамтылған әмбебап регистрлер де болады. Осындай әмбебап регистрдің мысалы ретінде 74194 аталымды төртразрядты регистрдің сызба белгілемесі келтірілген (1.25 сурет). Оның шықпаларының қызметі 1.15 кестеде көрсетілген.



1.25 сурет

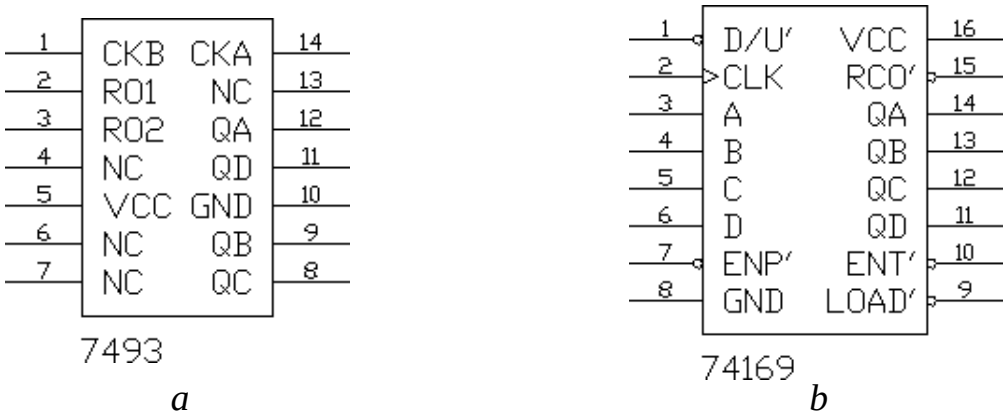
1.15 к е с т е

CLR’	S1	S0	CLK	SL	SR	D	C	B	A	QD	QC	QB	QA
0	*	*	*	*	*	*	*	*	*	0	0	0	0
1	0	0	⎍	*	*	*	*	*	*	QD0	QC0	QB0	QA0
	0	1		*	*	*	*	*	*	QC0	QB0	QA0	SR
	1	0		*	*	*	*	*	*	SL	QD0	QC0	QB0
	1	1		*	*	*	*	*	*	D	C	B	A

1.4.3 Санауыштар.

Санауыш – импульстердің әрбір сәйкесті ауытқымасында құрамындағы санды бірге өзгертіп отыратын құрылғы. Құрамындағы санның өзертілу бағытына байланысты тура, теріс және әмбебап санауыштар болады. Санауыштар Т-триггерлердің негізінде құрылады.

Қарапайым санауыштың мысалы ретінде Electronics Workbench бағдарламасының элементтер қорындағы 7493 аталымды төртразрядты санауыш микросұлбасының белгілемесі көрсетілген (1.24, *а* сурет). Оның құрамында орналастырылған жеке бірразрядты (кірісі – CKA, шығысы – QA) және үшразрядты (кірісі – CKB, шығыстары – QD, QC, QB) санауыштарды жеке түрінде де, оларды бір-біріне жалғап, төртразрядты санауыш ретінде де пайдалануға болады.



а *b*

1.26 сурет

7493 микросұлбасының шықпаларының қызметі 1.16 кестеде көрсетілген.

1.16 к е с т е

RO1	RO2	CLK	QD	QC	QB	QA
1	1	*	0	0	0	0
0	*	⌐	+1			
*	0					

Әмбебап санауыштың мысалына ретінде 74169 аталымды төртразрядты санауыштың микросұлбасы жатады (4.26, *b* сурет). Оның шықпаларының қызметі 1.17 кестеде көрсетілген.

1.17 к е с т е

ENP’	ENT’	D/U’	CLK	LOAD’	D	C	B	A	QD	QC	QB	QA	RC0’
0	0	*	*	0	*	*	*	*	D	C	B	A	
0	0	0	⌐	1	*	*	*	*	–1				M: 1→0
0	0	1		1	*	*	*	*	+1				
1	*	*	*	*	*	*	*	*	QD0	QC0	QB0	QA0	
*	1	*	*	*	*	*	*	*					

2 Цифрлық құрылғылардың моделдері

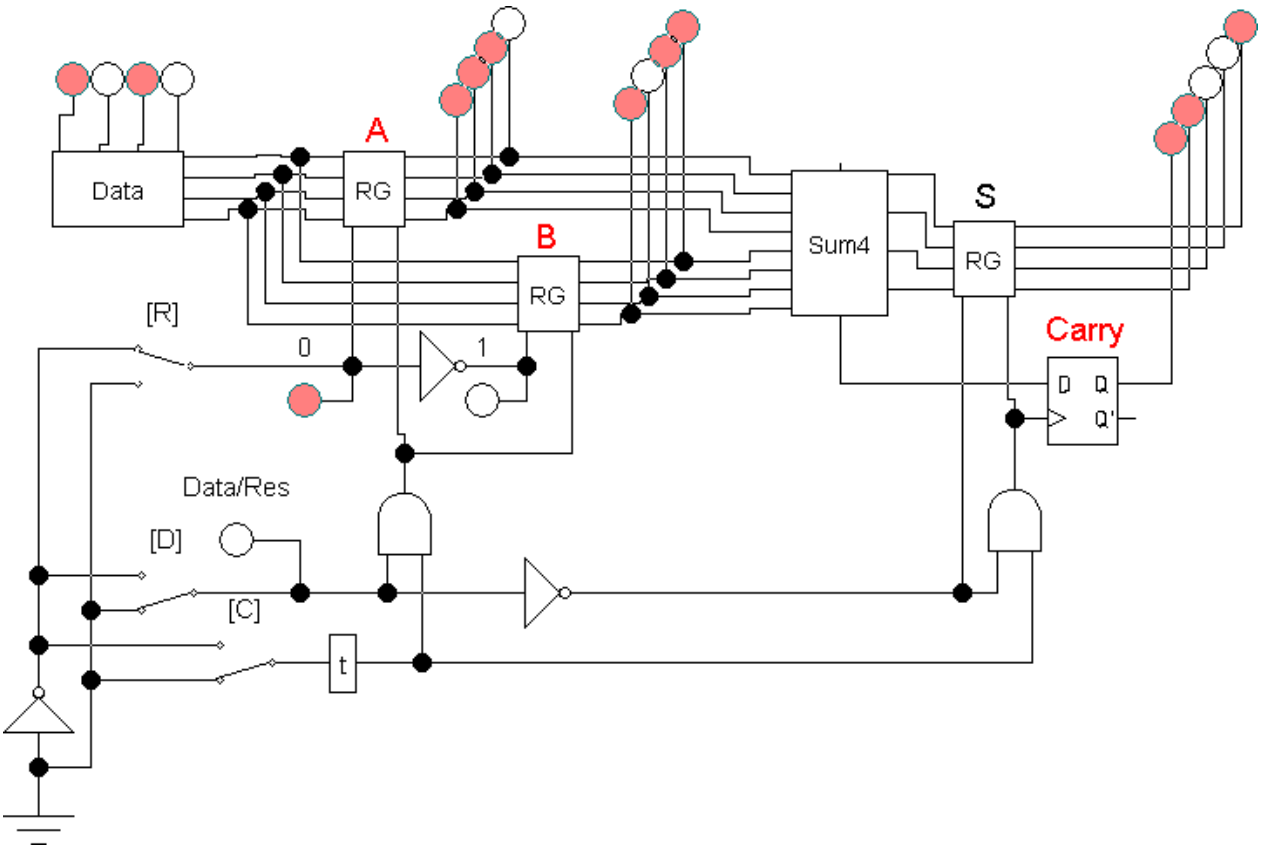
Цифрлық құрылғылардың құрылымын логикалық және схемотехникалық тұрғыдан құру мүмкіншілігін көрсету мақсатында схемотехникалық моделдеу жұмысын жүргізуге арналған Electronics Workbench бағдарламалы ортасында алдында қарастырылған қалыпты қиыстырма және тізбектеме арқылы құрылған, нақтылы алгоритмдерді жүзеге асырушы бірнеше цифрлық құралымның тәжірибелік моделдерін суреттелік.

Бұдан әрі келтірілген және суреттелетін қосуыштардың кейбір түрлерінің моделдері төртразрядты деректермен жұмыс істейтіндей етіп құрылған. Қажетті жағдайда оларды разряд саны жоғарырақ деректермен (мысалы, сегізразрядты деректермен) істейтіндей етіп жаңарту қиын емес.

2.1 Тізбелі ауыспалы қосу құрылғысы

Бұл тарауда төртразрядты екі санның қосылуын жүзеге асырушы құрылғының моделі суреттеледі (2.1 сурет).

Бұл құрылғыда қосылуға арналған бастапқы деректерді қалыптастыру үшін жиілігі 1 Гц генератордан істеуші төртразрядты санауыштың 7493 микросұлбасының негізінде Data блогы құрылған.



2.1 сурет

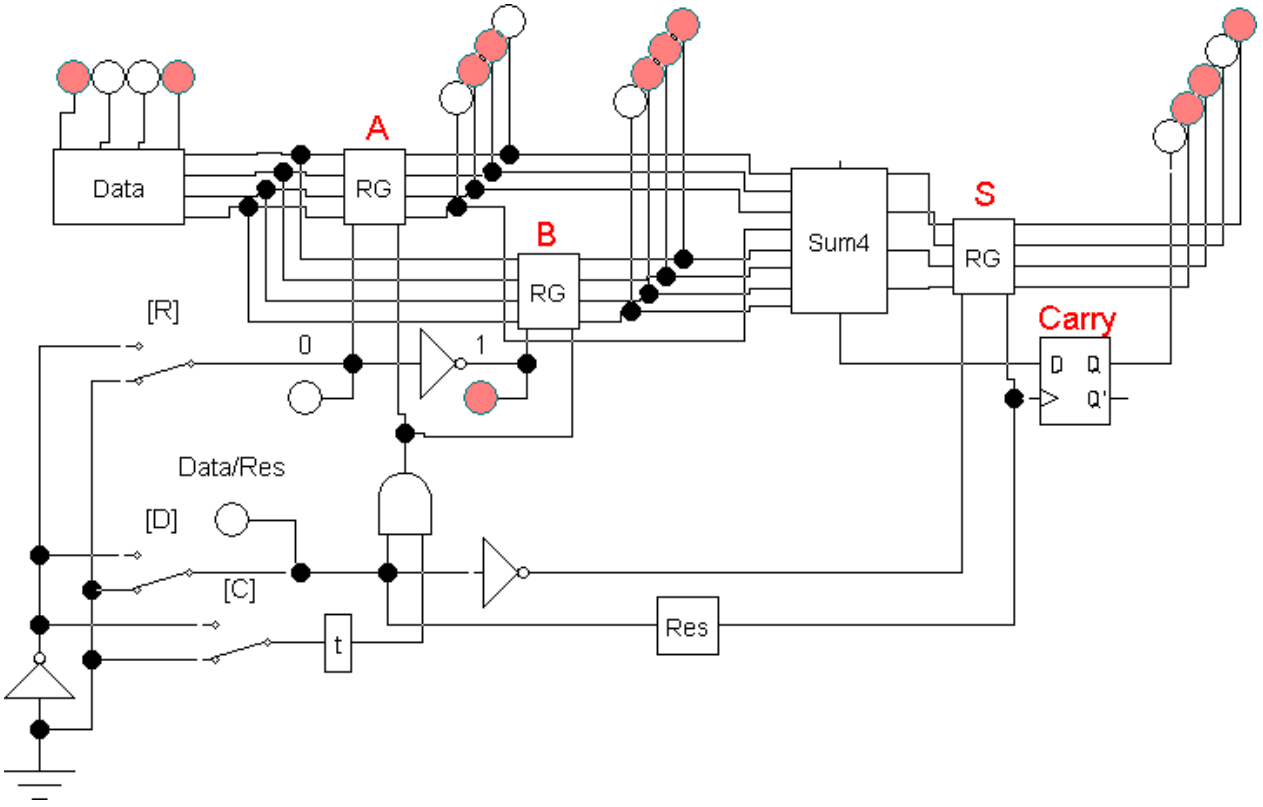
Data блогының индикаторларында 0 – 15 арасындағы сандардың коды 1 секунд сайын (генератордың қойылған жиілігіне сәйкесті) тізбелі түрде шығарылып тұрады. Қажетті сан кодының шығуын күтіп, оны кіріс деректерінің қабылдағышына жазып алу ғана қалады.

Біздің құрылғыда дерек қабылдағыштары ретінде, 74194 регистр микросұлбасының негізінде құрылған, сәйкесті RG блоктары пайдаланылған. Дерек жазуға қажетті регистрді таңдау R ауыстырғышына сәйкесті түйменің жағдайын өзгерту арқылы жүзеге асырылады. Бұл кезде дерек қабылдауға дайындалған регистр сәйкесті индикатордың жарықтануымен көрсетіледі.

Қарастылушы құрылғының екі жұмыс режимі бар, олар D ауыстырғышының жағдайымен қойылады: бастапқы деректер енгізу режимі (D = 1) және дайындалған деректердің қосылуын жүзеге асыру режимі (D = 0).

Кіріс регистрлеріне (A және B) деректер енгізу, қосылым нәтижесін S регистріне және тасымалды Carry триггеріне жазу жалғыз C түймесімен жүзеге асырылады. Бұл кезде XNOR элементінің негізінде құрылған t блогы (1.6 сурет) строб сигналын қалыптастырады да, оның оң ауытқымасымен регистрлерге дерек жазу жүзеге асырылады.

Төртразрядты қосуыш сұлбасы бірразрядты қосуыштар негізінде құрылған.



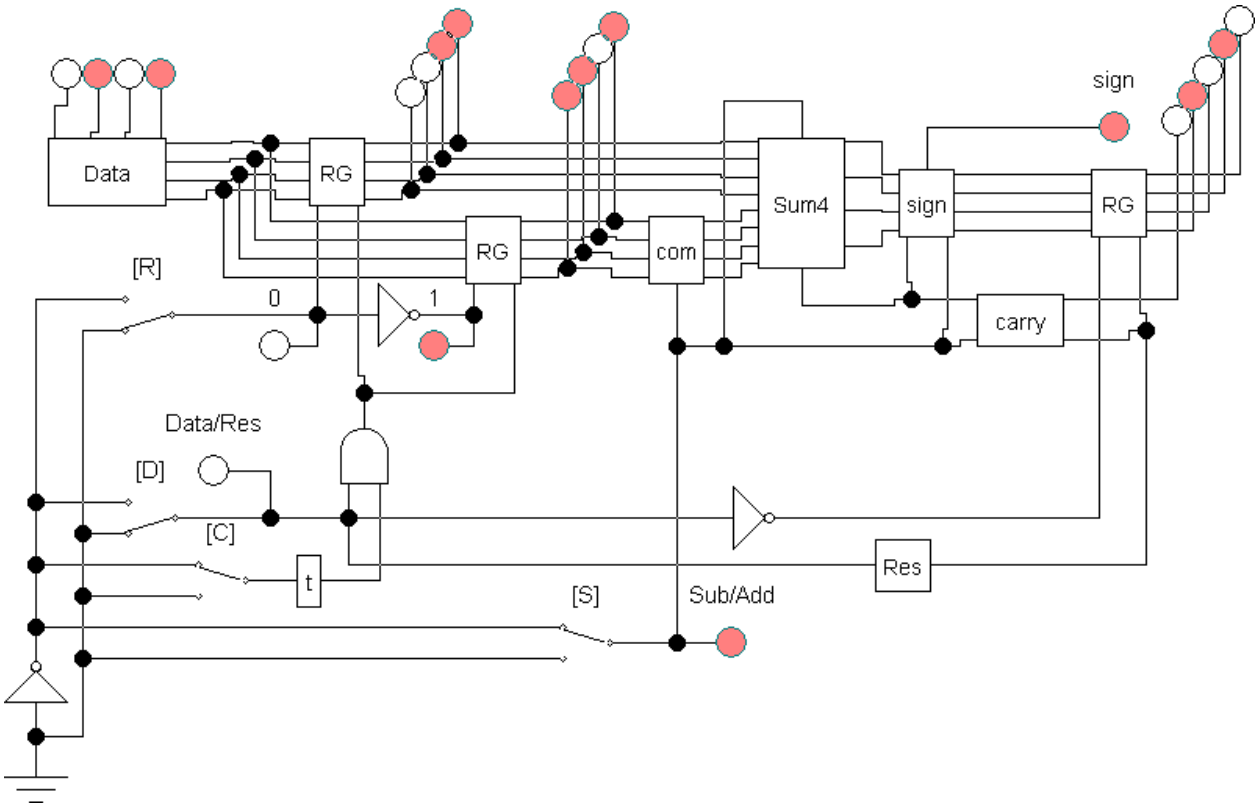
2.2 сурет

Алдыңғы сұлбаның жаңартылған түрінде (2.2 сурет) қосылым нәтижесі мен тасымалды жазып алу құрылғыны жазу режиміне ауыстыру арқылы, яғни D түймесінің 1 жағдайынан 0 жағдайына ауыстырылуы арқылы жүзеге

асырылады. Сәйкесті жазу стробы, 1.4, *е* суреттегі сұлбамен құрылған, Res блогы арқылы қалыптастырылады.

2.2 Алу-қосу құрылғысы

Сұлба жаңартылымын одан әрі жалғастыралық, нақтылы айтқанда, дайындалған сандарды қосып та, алып та бере алатын құрылғы құралық. Осындай жаңарту нәтижесі 2,3 суретте келтірілген.



2.3 сурет

Келтірілген құрылғыда орындалатын операция S түймесінің жағдайымен анықталады: S = 0 кезінде – қосу, ал S = 1 кезінде – алу.

Цифрлық құрылғыда арифметикалық операциялардың қосымша код түрінде көрсетілген деректермен орындалатындықтан, алындының қосымша кодын қалыптастыру қажеттігі туады. Теріс санның қосымша кодын қалыптастыруды оның тура кодының рарядтарының мәндерін терістеп, алынған терістелген кодқа бірлік мәнін қосу арқылы жүзеге асыруға болады. Қарастырылушы құрылғыда алынды кодының рарядтарының мәндерін терістеу, XOR элементінің негізінде құрылған, com блогында, ал терістелген кодқа бірлік мәнін қосу қосуыштың өзінде жүзеге асырылады.

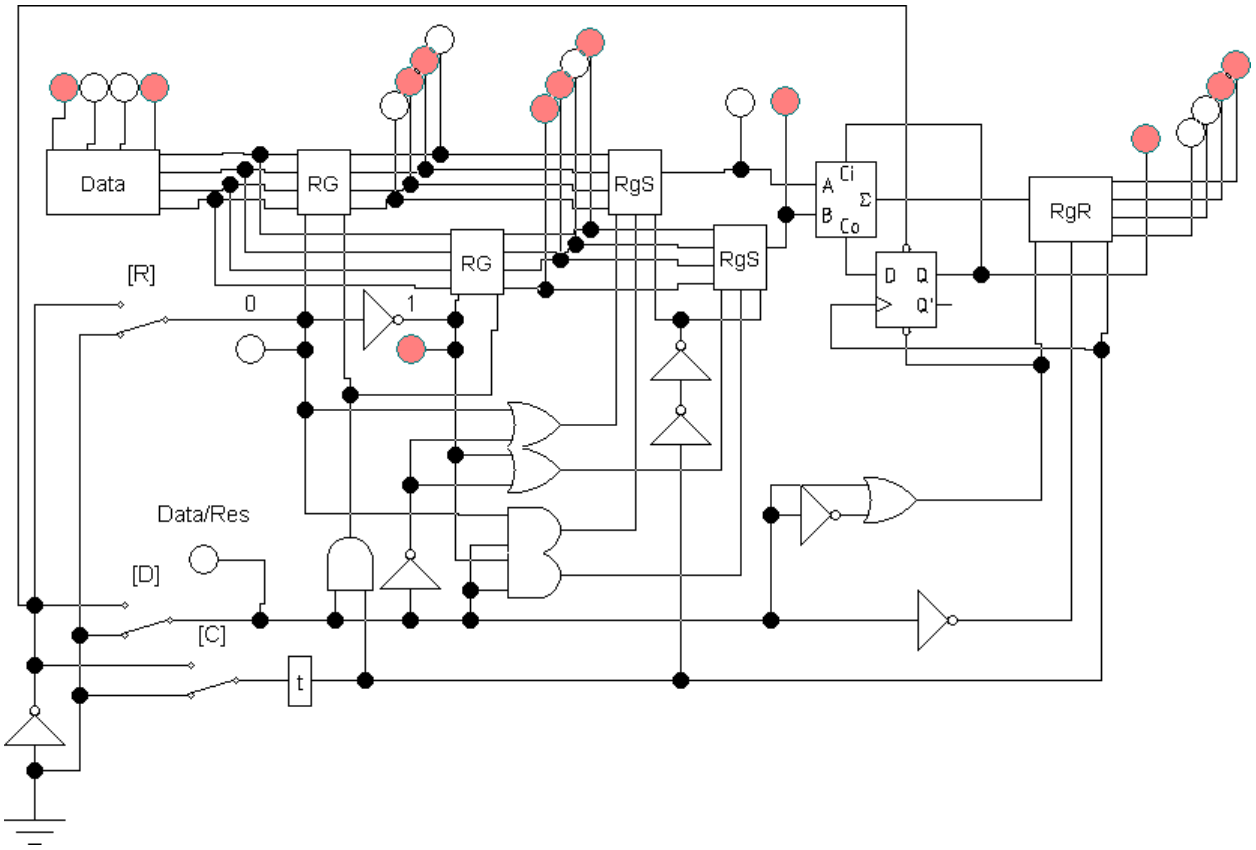
Сан кодын қосымша кодқа түрлендіру қажеттігі шығарылған нәтиженің теріс сан болған кезінде де туады. Бұл жағдайдағы сәйкесті түрлендірімдер sign блогында жүзеге асырылады. Нәтиженің теріс таңбасы да осы блокта қойылады.

2.3 Тізбелі операндтарға арналған қосу құрылғысы

Алдыңғы қарастырылған қосу құрылғыларының сұлбалары төрт бірразрядты қосуыштар арқылы құрылған болатын. Тізбелі операндтарға арналған қосу құрылғысында бір ғана бірразрядты қосуыш пайдаланылады да, қосу операциясы сан кодтарының разрядтары бойынша, кіші разрядтан бастап, тізбелеп қосу арқылы жүзеге асырылады (2.4 сурет).

Тізбелі қосу құрылғысының сұлбасының құрамында бірразрядты қосуышпен қатар қосылғыштардың (екі RgS) және қосындының (RgR) ығыстыру регистрлері мен разряд аралық тасымал сақтауға арналған триггер орналастырылған.

Бастапқы деректердің сәйкесті RG регистрлерінен RgS регистрлеріне қайта жазылуы екі NOT элементі арқылы тудырылатын 20 ns шамасындағы кідіріспен жүзеге асырылады. RgS регистрлерінің жұмыс режимдері (жазу, сақтау, ығыстыру) бұл регистрлердің S0 және S1 кірістеріне жалғанған қиыстырма сұлбалар арқылы R және D ауыстырғыштарының жағдайына сәйкесті қойылады. Құрылғының жұмыс режимін қосу режиміне ауыстырған кезде нәтиже регистрі RgR мен тасымал триггері тазартылады.



2.4 сурет

Разрядтар бойынша қосу С түймесін басу арқылы қалыптастырылатын, төрт тактілік сигналмен жүзеге асырылады. Сәйкесті индикаторлар разрядтар бойынша қосылу барысын көрнекі бақылауға мүмкіндік береді.

2.4 Автоматты режимдегі тізбелі қосу құрылғысы

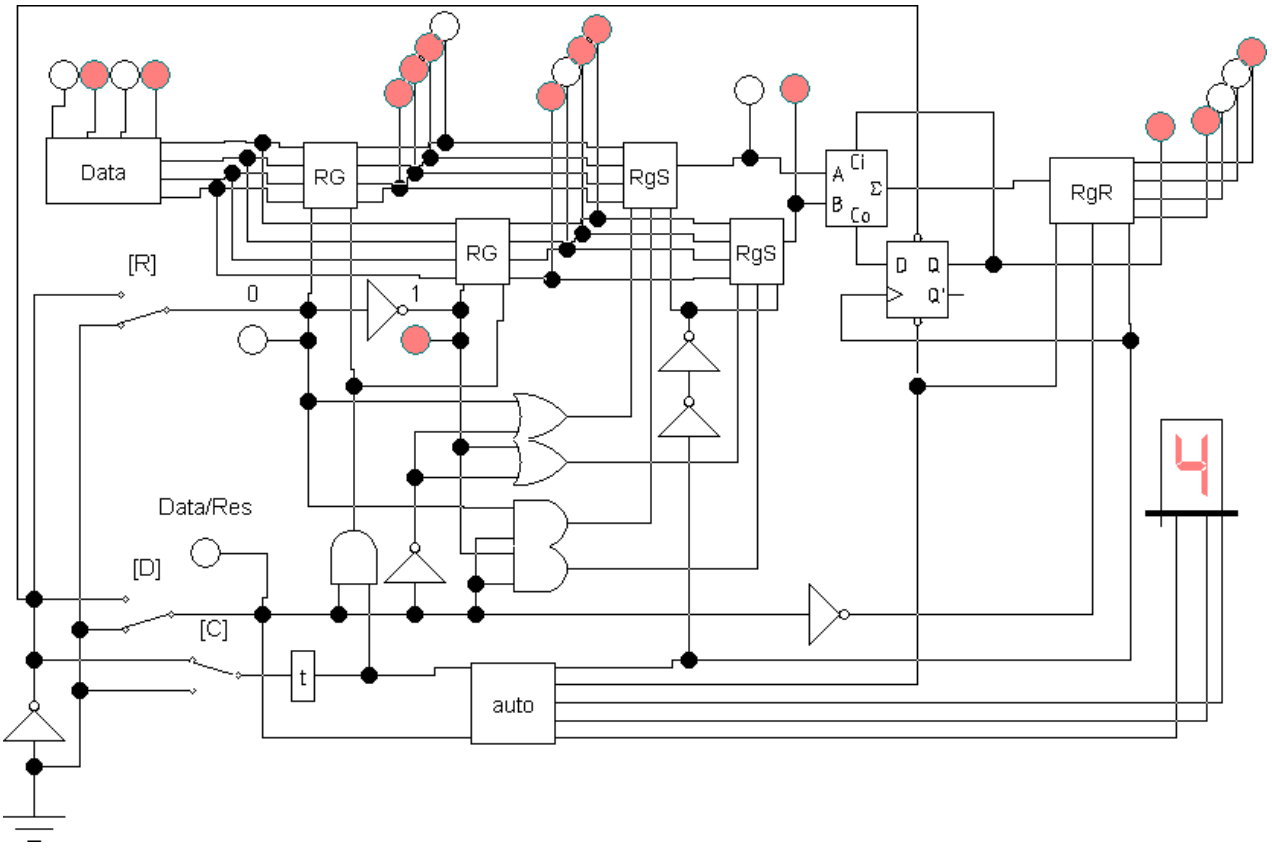
Алдыңғы тараудағы қосу құрылғысында разрядтар бойынша қосу С түймесін басу арқылы қалыптастырылатын, төрт тактілік сигналмен жүзеге асырылатынды. Әрине, бұл операциялардың автоматты режимде орындалуы, яғни құрылғыны қосу режиміне ауыстыру кезінде (D ауыстырғышын 1 жағдайынан 0 жағдайына өзгерту арқылы) осы тактілік сигналдарды қалыптастырушы автоматтың іске қосылғаны ыңғайлы болар еді. 2.5 суретте тізбелі қосуыштың сәйкесті жаңартылған сұлбасы келтірілген.

Құрылғы құрамына енгізілген, 7493 санауыш микросұлбасының негізінде құрылған auto блогы (2.6 сурет), келесі қызметтер атқарады:

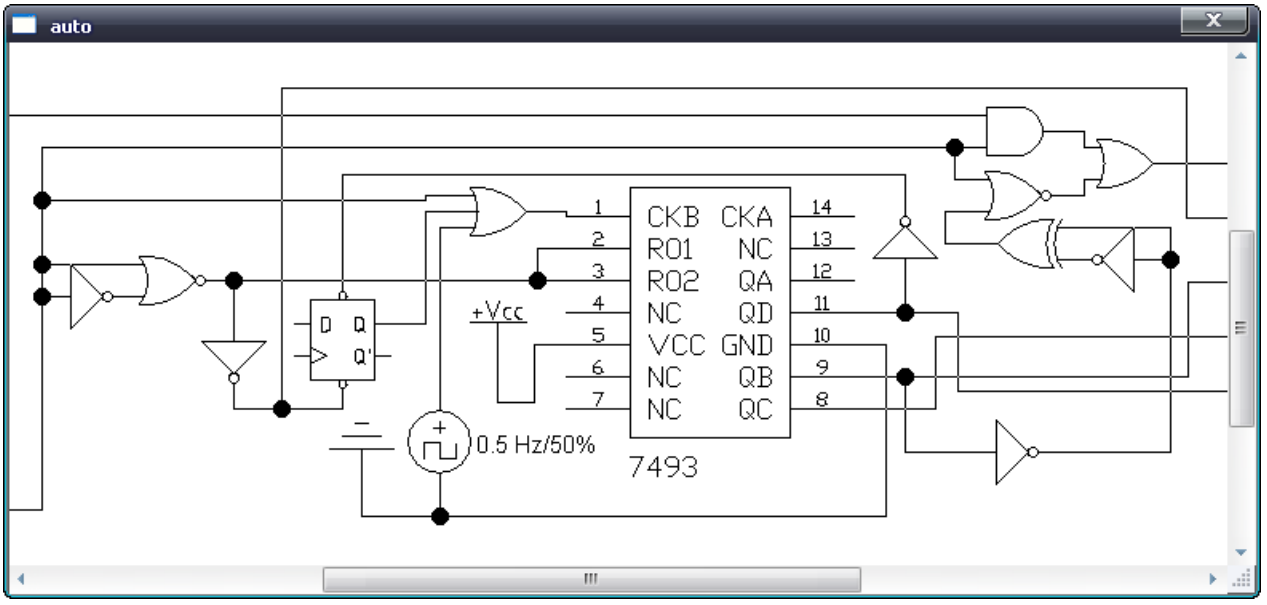
– деректер дайындау режимінде жазу стробтарын RgS блоктарына (оларға RG блоктарындағы деректердің қайта жазылуын жүзеге асыру үшін) жібереді;

– қосу режиміне ауыстырылу мезетінде қалыптастырылатын сигнал (ол бұрын нәтиже регистрі RgR мен тасымал триггерінт азартуға ғана пайдаланылғанды) auto блогының құрамындағы триггерді нөлдік жағдайға қояды да, генератор импульстерінің санауышқа өтуіне мүмкіндік береді;

– сосын төрт тактілік сигнал (стробтар) қалыптастырылады да, одан кейін санауыш жұмысы тоқтатылады (төртінші строб қалыптасқаннан кейін аздаған кідіріспен).



2.5 сурет



2.6 сурет

3 Дерек сақтау құрылғылары

Дерек сақтау құрылғылары (өзгеше айтқанда, жады құрылғылары), олардың орналастырылған цифрлық жүйелерінде, ақпарат сақтау қызметін атқарады. Цифрлық жүйелердің құрамына, олардың атқаратын қызметіне байланысты жады құрылғыларының бірнеше түрі енгізілуі мүмкін, олар:

– микропроцессорлардың құрама бөлігі ретіндегі регистрлік жады;

– цифрлық жүйенің тез әрекеттілігін көтеруге мүмкіндік беретін кэш-жады;

– процессормен тікелей дерек алмастыру режимінде жұмыс атқарушы және онымен тезәрекеттілігі жағынан келістірілген негізгі жады (қызмет жадысы, тұрақты жады);

– үлкен көлемді ақпарат сақтауға арналған сыртқы жады және т.б.

Жады құрылымдарының жұмыс мүмкіндігін сипаттаушы негізгі параметрлері:

– ақпараттық сыйымдылығы – сақталатын ақпараттың ең жоғарғы көлемі. Ол бит, байт немесе бірнеше байттан тұратын, сөз санымен сипатталады. Бұл параметр, көбінесе сақталатын сөз санының олардың разряд санына көбейтіндісі түрінде сипатталады;

– тезәрекеттілігі – оқу, жазу операцияларының және оқу/жазу циклінің ұзақтығымен бағаланады. Оқу (немесе жазу) уақыты – басқару (оқу/жазу) сигналының берілген мезетінен сәйкесті операцияның орындалып біткеніне дейінгі уақыт аралығы. Оқу/жазу циклінің ұзақтығы – тізбелі жүргізілетін операциялардың минималды уақыт аралығы.

Қазіргі заманғы сөз тобымен (бумасымен) жұмыс істейтін жады құрылымдарында айтылған дәстүрлі параметрлермен қатар бастапқы

байланыс уақыты (Latency) және буманың келесі сөздерінің жіберілім қарқыны (Bandwidth) аталымды жаңа параметрлер енгізілген.

Келтірілген динамикалық параметрлермен қатар, бірнеше басқару сигналының өзара орналасымы мен олардың ұзақтығын анықтаушы режимдік параметрлер де бар.

Жадылық құрылғылардың жұмысы келесі сигналдармен басқарылады:

– A (Address) – адрес, оның разрядтылығы (n) жады құрылымының максималды ұяшық санымен ($N=2^n$) анықталады;

– CS (Chip Select) немесе CE (Chip Enable) – микросұлбаны таңдау немесе іске қосу сигналы;

– R/W (Read/Write) – сәйкесті операцияны орындату сигналы;

– DI (Data Input) және DO (Data Output) – кіріс және шығыс деректерінің сигналдары жүретін желілер. Кейбір жады құрылымдарында олар біріктірілген түрінде пайдаланылады.

Деректерге шығу тәсіліне қарай жады құрылғыларының сілтеу арқылы шығарылымды, тізбекті шығарылымды және танымалды шығарылымды түрлері ажыратылады.

Сілтеу арқылы шығарылым кезінде жады құрылғысының адрестік кірістеріндегі код дерек адмастырылуы жүргізілетін ұяшықты көрсетеді. Бұндай жады құрылғылары аса дамығандығына байланысты жадының өзге түрлері осылардың негізінде құрылады.

Сілтеу арқылы шығарылымды жады құрылғылары өз кезегінде RAM (Random Access Memory) және ROM (Read Only Memory) түрлеріне бөлінеді.

RAM түрлі жады құрылғыларда кезекті бағдарламаның орындалу кезіндегі алмасуға қатысты кез келген уақытта өзгеретін деректер сақталады. Олардың құрамындағы жады элементтері қорек көзіне тәуелді болады.

RAM түрлі жады құрылғылары құрамындағы жады элементтерінің түріне қарай статикалық (SRAM) және динамикалық (DRAM) құрылымдарына бөлінеді. SRAM құрылымдарында жады элементтері ретінде триггерлер пайдаланылады, сондықтан олардың тезәрекеттілігі жоғары болады, ал DRAM құрылымдарында деректер МТШ-құрылымның элементтері арқылы құрылған конденсаторлардың зарядтары түрінде сақталады. Конденсаторлардың өздігінен зарядсыздануы деректерді жояды, сондықтан олардың қайта-қайта (әрбір миллисекунд сайын) қалпына келтіріліп отырылуы керек. Бірақ динамикалық жады элементтерінің орналастырылым тығыздығы статикалық құрылыммен салыстырғанда бірнеше есе асып түседі.

Динамикалық RAM түріндегі жады құрылымдарының ақпараттық сыйымдылығы жоғары болады және арзанға түседі, сондықтан олар есептеу машиналарының негізгі жадысы ретінде пайдаланылады. Статикалық RAM түріндегі жады құрылымдары динамикалы құрылымдармен салыстырғанда бірнеше есе қымбатқа түседі (жады элементтерінің күрделілігінен) және олардың ақпараттық сыйымдылығы да сонша есе кем болады. Олардың артықшылығы – тезәрекеттілігінде, осыған байланысты қолданылым аймағы – кэш-жадысының сұлбалары.

ROM түрлі жады құрылымдарының кейбір түрінде ақпарат ешқашан өзгертілмейді, ал кейбірінде өзгертуге болады, бірақ ол құрылғының жұмыс режимінде емес, арнайы режимде ғана жүзеге асырылады.

ROMтүрлі ЖҚ-лар бағдарлану тәсіліне (және орнына) байланысты бірнеше түрде шығарылады, олар: ROM(M), PROM, EPROM және EEPROM.

ROM(M) түріндегі құрылымдар жинам технологиясымен өндірісте шығарылу кезінде арнайы маска арқылы бағдарланады да, одан кейін ондағы ақпарат ешқашан өзгертілмейді.

PROM (Programmable ROM) түріндегі құрылымдарды пайдаланушының өзі бағдарлайды (яғни оған қажетті ақпаратты енгізеді), одан кейін бұл құрылым ROM(M) сияқты тұрақты жады ретінде пайдаланылады.

EPROM (Electrically Programmable ROM) және EEPROM (Electrically Erasable Programmable ROM) түріндегі құрылымдардағы ақпаратты өшіріп, одан кейін оларды қайтадан бағдарлауға болады. EPROM құрылымдарындағы ақпарат ультракүлгін сәулесімен, ал EEPROM құрылымдарындағы ақпарат электр сигналдарымен өшіріледі. Оларды бағдарлау зертханалық жағдайда арнайы бағдарлаушы құрылғылар арқылы жүзеге асырылады.

Тізбекті шығарылымды жады құрылымдарында жазылған деректер кезек құрады. Кезектегі сөздердің оқылуы олардың жазылым тәртібімен немесе керісінше тәртіппен жүзеге асырылады. Сөздердің тура шығарылым тәртібі FIFO (First In – First Out) буферлері мен файлды және циклді ЖҚ-ларда қолданылады. Сөздердің кері тәртіппен оқылуы LIFO (Last In – First Out) буферлері ретінде істейтін стек түріндегі жады құрылымдарында қолданылады.

Танымалды шығарылымды жады құрылымдарындағы ақпаратты іздеу оның жадыдағы орналасымы арқылы емес, сөз құрамына енгізілген арнайы танымал белгісі (tag) арқылы жүзеге асырылады. Олар, негізінде, үлкен көлемді цифрлық жүйелердің (мысалы, есептеу машиналарының) құрамындағы кэш-жадыларда пайдаланылады.

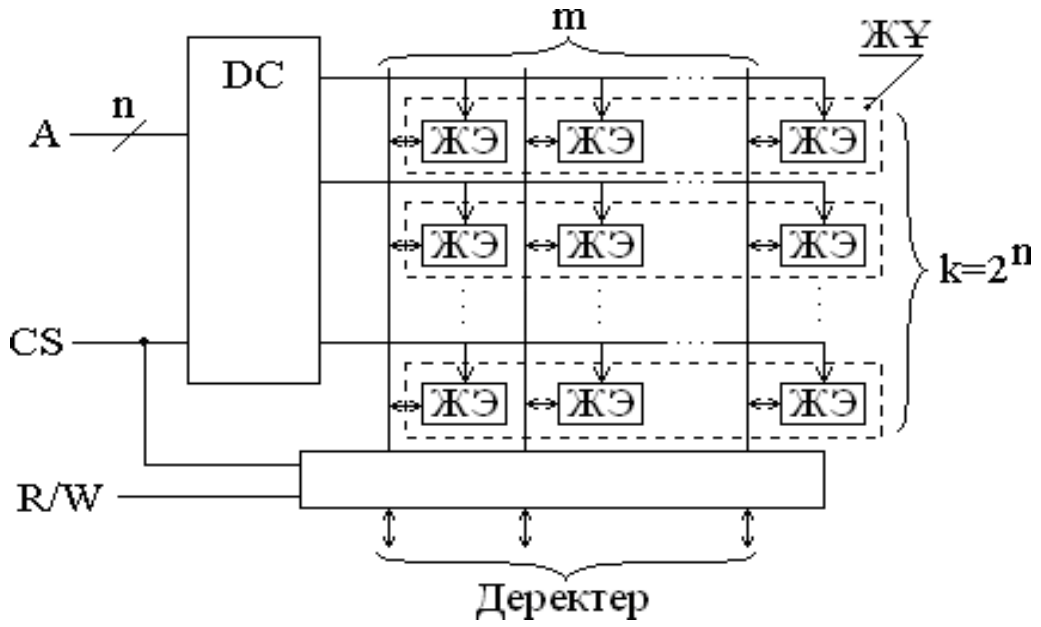
3.1 RAM түріндегі жады құрылғыларының құрылымдары

Статикалы RAM түрлі жады құрылғылары 2D, 3D және 2DM құрылымымен құрылады.

2D құрылымында (3.1 сурет) m-разрядты жады ұяшықтары (MC, Memory Cell) дешифратордың сәйкесті шығыстарына қосылады, яғни олар бір жол бойында орналастырылады (суретте – вертикаль бойында), сондықтан бұндай құрылымдарды сызықты құрылым деп атауға болады.

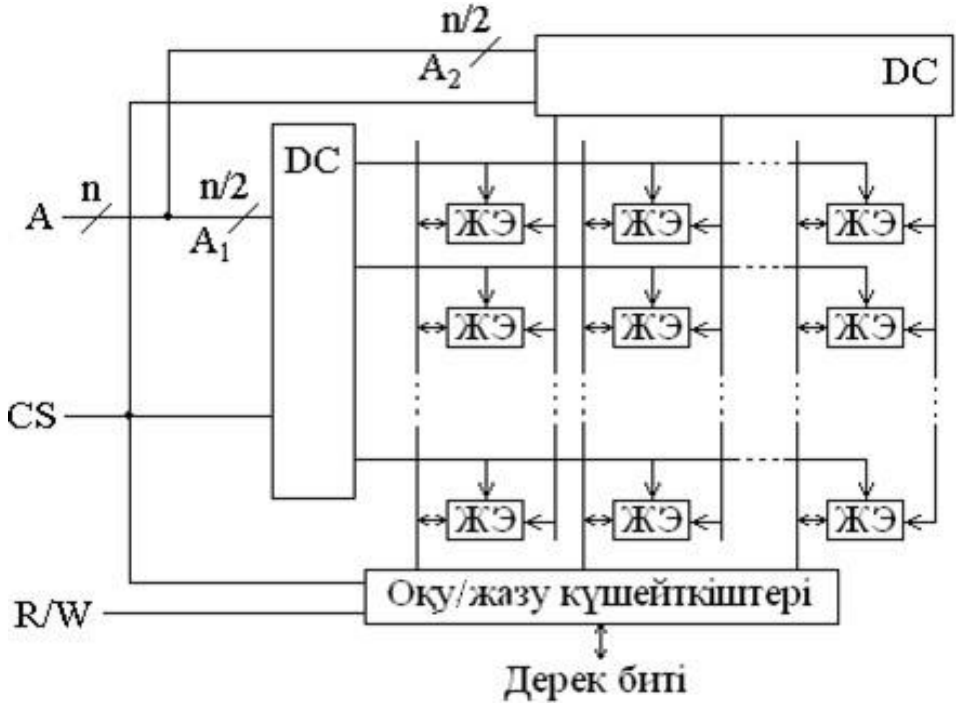
Адрес кодының дешифраторы (DC), рұқсат сигналы CS (Chip Select) берілген кезде шығыс жолдарының біреуін жандандырып, таңдалған жолдағы ұяшыққа, демек ондағы сөзге қол жеткізеді. Матрица бағаналарының сәйкесті элементтері вертикаль жолға, яғни ішкі дерек жолына (разряд жолына, оқу/жазу жолына) жалғанған. Бағана элементтерінде сөздердің аттас биттері

сақталған. Сөздің жіберілім бағыты R/W (Read/Write) сигналының әрекетіне байланысты оқу/жазу күшейткіштерімен анықталады.



3.1 сурет

3D құрылымы (матрицалы құрылым) қоскоординаталы талғам арқылы адрес дешифраторын жеңілдетуге мүмкіндік береді (3.2 суретті қара).



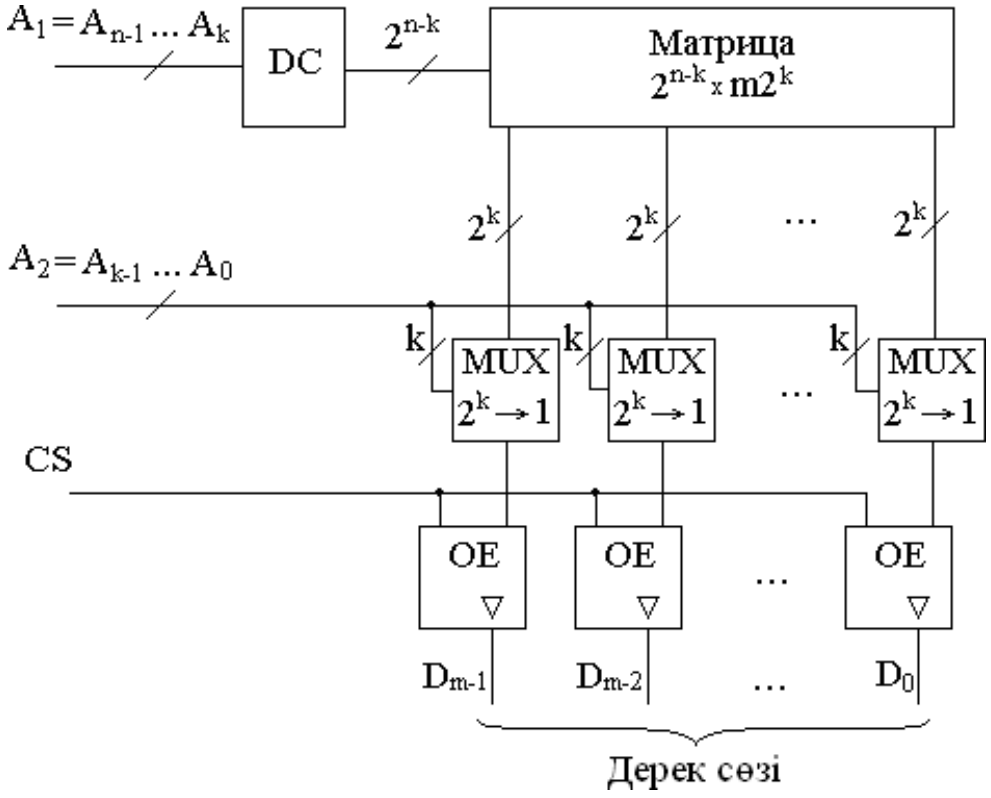
3.2 сурет

Бұнда n разрядты адрес коды екіге бөлінеді де, олардың әрбіреуі жеке ажыратылады. Жандандырылған жолдардың қиылысындағы жады элементі

(немесе ұяшығы) таңдалады. Бұндай қиылыстар саны $2^{n/2}$ x $2^{n/2}$ = 2^n. Екі дешифратордың шығыстарының толық саны $2^{n/2} + 2^{n/2} = 2^{n/2+1}$ болады, ал бұл 2D құрылымдағы дешифратордың шығыс санынан (2^n) едәуір кем.

Үлкен құрылымдардың (мысалы, компьютерлердің) жады жүйелерін ұйымдастыруға арналған 2DM (модификацияланған 2D) құрылымында 2D және 3D құрылымдарының екеуінің де артықшылық жақтары қамтылған – адрес ажыратылуы жеңілдетілген және екікоординаталы талғанымды жады элементтерінің қажеттігі жоқ.

2DM құрылымының жады элементтерінің матрицасы (3.3 сурет) 2D құрылымының сипатындай: дешифратордың жандандырылған шығысы толық бір жолды таңдайды. Бірақ 2D құрылымынан ерекше, бұнда жол ұзындығы сақталушы сөздердің разряд санындай емес, одан бірнеше есе асып түседі.



3.3 сурет

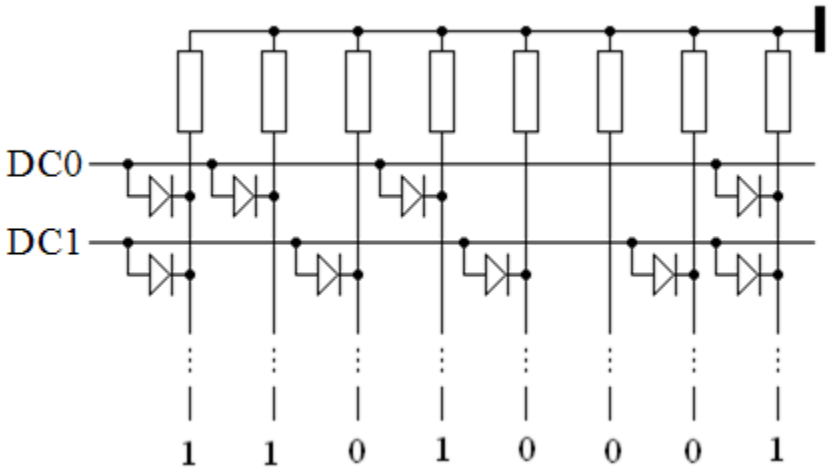
Жолдардың біреуін таңдауға адрес кодының A_{n-1}... A_k бөлігі пайдаланылады. Адрестің қалған разрядтары (от A_{k-1}... A_0) таңдалған жолдағы көптеген сөздердің қажеттісін мультиплексор арқылы бөліп шығаруға пайдаланылады. Жол ұзындығы – m^{2k}, бұндағы m – сақталушы сөздердің разряд саны. Ұзақтығы 2^k жолдың әрбір "кесіндісінен" мультиплексор бір бит таңдайды. Мультиплексорлардың шығыстарында шығыс сөзі қалыптасады. Үш жағдайлы басқарылымды буферлердің OE кірістеріне түсетін CS сигналының рұқсатымен шығыс сөзі сыртқы желіге жіберіледі.

3.2 ROM түріндегі жады құрылғыларының элементтері

ROM (Read Only Memory) түрлі жадылық құрылғылардың кейбіреуіндегі ақпарат (ROM(M) және PROM түрлі ЖҚ) ешқашан өзгермейді, ал кейбіреуінде (EPROM және EEPROM түрлі ЖҚ) өзгертіледі, бірақ ол құрылғының жұмыс режимінде емес, арнайы режимде жүзеге асырылады.

ROM(M) түріндегі құрылымдар жинамды технологиямен олардың өндірісте шығарылу кезінде арнайы маскалар арқылы бағдарланады да, одан кейін ақпарат ешқашан өзгертілмейді. Оларды бағдарлау жады элементтерінің матрицасының горизонталь және вертикаль жолдарының арасында байланыс элементтерінің орналастырылуы арқылы жүзеге асырылады. Байланыс элементтері ретінде диодтардың, биполярлы транзисторлардың, МОШ-транзисторлардың және т.б. пайдаланылуы мүмкін.

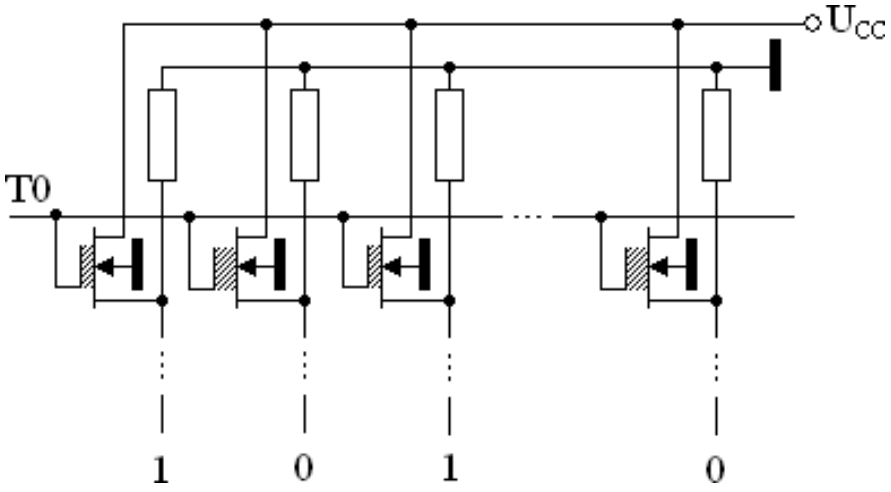
Диодты ROM(M) матрицасында (3.4 сурет) горизонталь жолдар – сөз талғам жолдары, ал вертикаль жолдар – оқылым жолдары. Оқылатын сөз диодтардың координат торының түйіндеріндегі орналасысымымен анықталады. Диодтың тұрған кезінде таңдалған горизонталь жолдың жоғары мәнді потенциалы сәйкесті вертикаль жолға жіберіледі де, сөздің осы разрядында логикалық бір сигналы шығады. Диодтың жоқ кезінде, вертикаль жол резистор арқылы жерге жалғанғандықтан, осы жолдың потенциалы нөлге жуық болады. Көрсетілген матрицадағы талғам жолы DC0 жандандырылғанда 11010001 сөзі оқылады (нөмірі нөл ұяшықта осы сөз сақталған). DC1 жолы жандандырылғанда 10101011 сөзі оқылады. Талғам жолдары – адрес дешифраторының шығыстары, әрбір адрес қиыстырымы дешифратордың сәйкесті шығысын жандандырады да, сілтенген ұяшықтағы сөз оқылады.



3.4 сурет

МТШ технологиясымен құрылған ROM(M) құрылымының матрицасының қиылыстарында орналастырылатын жады элементі ретінде МТШ-транзисторлар қолданылады (3.5 сурет). Құрылымның сәйкесті қиылысындағы нөл мәнін сақтаушы МТШ-транзисторларының арнасы мен

жапқышының арасындағы тотық қабатының қалыңдығы ұлғайтылады да, ол транзистордың шекті ашылым кернеуін өсіреді, яғни ол ашылмайтын жағдайға келтіріледі, ал бір мәнін сақтаушы транзисторлардың тотық қабатының қалыңдығы оның дешифратордан түсетін талғам сигналымен ашылатындай мөлшерге қойылады.



3.5 сурет

ROM(M) түрлі жады құрылғыларының жады элементтері өте аз орын алады, демек сәйкесті микросұлбалардың жиналым деңгейі жоғары болады. Бірақ жиналымды технологиялық тәсілмен бағдарлауға арналған маскаларды жобалау мен жасауға кететін қаражат аса жоғары болады. Бұндай қаражат олардың кең қолданылатын жағдайында ғана ақталады. Бұдан олардың қолданылым аймағы да анықталады, олар – тұтынушылардың көпшілігіне қажетті, стандартты ақпарат сақтау. Мысалы, маскалы бағдарламалы жады құрылғыларында алфавит (орысша және латынша) символдарының, қалыпты функциялардың (синустың, квадраттық функциялардың және т.б.) кестелері, стандартты бағдарламалар және с.с. кең қолданылымды ақпараттың "тігілімдері[1]" болады.

PROM (Programmable ROM) түрлі жады құрылғыларының микросұлбалары арнайы жалғамаларды жою немесе оларды құру арқылы бағдарланады. Микросұлбаның бастапқы дайындалымында барлық жалғамалар болады (немесе ешқандай жалғама болмайды). Бағдарлаудан кейін олардың қажеттілері ғана қалады (немесе құрылады).

Жойылымды жалғамалы жады құрылғыларының бастапқы микросұлбасында барлық жалғама орналастырылған (яғни жады матрицасының барлық қиылысында лог.1 тұрады). Бағдарлау кезінде олардың кейбірі амплитудасы мен ұзақтығы жеткілікті мөлшердегі ток импульстарымен еріту арқылы жойылады (яғни жады матрицасының сәйкесті

[1]"Тігілім" – тұрақты жадының құрамын суреттеуге пайдаланылатын, қалыптасқан термин.

қиылыстарына лог.0 жазылады). Ерітілме жалғамалар диодтардың немесе транзисторлардың электродтарына қосылады (3.6, *а*-сурет).

Құрылымды жалғамалы жады құрылғыларының бастапқы микросұлбасының құрамындағы жады матрицасының барлық қиылысында қарсы қосылымды диод түріндегі құрылым орналастырылған (3.6, *b*-суретті қара). Қосдиодты құрылымның бастапқы жағдайдағы кедергісі жоғары болады да, бұл қиылыстардың ажыратылған жағдайына, яғни онда лог.0 сақталған элемент тұрғанына сәйкес келеді. Нақтылы қиылыстарды лог.1 жағдайына келтіру үшін, яғни онда бір жазу үшін, бұндағы диодтарға жоғары мәнді кернеу беру арқылы кері бағытта қосылған диод тесіліп, бұл қиылыс тұйықталады да, онда ток жүргізуші жалғама туады.

а *b*

3.6 сурет

Қайта бағдарламалы EPROM және EEPROM түрлі жады құрылғыларында ескі информацияны өшіріп, оның орнына жаңа ақпарат жазу мүмкіндігі қамтылған. Бұл операциялар (өшіру және жазу) олардың жұмыс режимінен шығарылған арнайы жағдайда жүргізіледі. Ақпараттың өшірілуі EPROM құрылымында ультракүлгін сәулесімен, ал EEPROM құрылымында электрлік тәсілмен жүзеге асырылады.

Қазіргі заманғы қайта бағдарламалы жады құрылғыларының жады элементтері ретінде жұмысы қалқыма жаппаның тудырылуына негізделген МНТШ-транзисторлар (Металл, Нитрид, Тотық, Шалаөткізгіш) мен ЗТИМТШ-транзисторлар (Зарядтардың Тасқынды Инжекциясына МТШ-транзисторлар) қолданылады [1].

4 Микропроцессорлық жүйелер

4.1 Микропроцессорлық жүйелердің құрылым принциптері

Микропроцессор (МП) – жалпы түрде микропроцессорлық жүйе (МПЖ) деп аталатын есептеу, бақылау-өлшеу және басқару жүйелерінің құрамындағы басты құрылғы. Микропроцессор ақпараттың өңделуін және оның жан-жақты жіберілімін бағдарлама арқылы басқарады. Қойылған мәселенің орындалу тәртібі сәйкесті бағдарламаға, яғни алынған микропроцессорға тән (яғни оның командалар жүйесіне кіретін) командалар тізбесінің орындалуы арқылы жүзеге асырылады.

Микропроцессорлық жүйенің құрамына микропроцессормен қатар жады құрылымдары және ақпаратты енгізу/шығару құрылғылары (сыртқы құрылғылар) кіреді.

Микропроцессорлық жүйелердің (МПЖ) құрылым негізіне үш принцип алынған, олар – магистральдық принцип, модульдік принцип және микробағдарламалы басқару принципі.

Магистральдық принцип МПЖ қызмет блоктарының арасындағы байланыс сипатын анықтайды. Жүйенің барлық блогы өзара жүйе магистралі (жүйе желісі) арқылы жалғанады және сол арқылы ақпарат алмасады.

Модульдік принцип жүйенің қызметі жағынан толықтай бітірілген жеке блоктар негізінде құрылатындығын сипаттайды. Әрбір модульдің оны іске қосатын рұқсат кірісі болады. Ондағы сигнал деңгейі модульдің үшінші (жоғары кедергілі) жағдайын, яғни оның жүйе желісіне қосылуын басқарады.

Микробағдарламалы басқару принципі қарапайым операциялардың – микрокомандалардың (ақпарат жіберілімінің, арифметикалық операциялардың, ығыстырылымдардың және т.б.) жүзеге асырылу мүмкіндіктерін сипаттайды.

Жүйе магистралінің құрамына негізгі үш ақпараттық желі кіреді, олар – адрес желісі АВ (Address Bus), дерек желісі DB (Data Bus) және басқарым желісі СВ (Control Bus).

Дерек желісі – жүйе модульдерінің арасында дерек алмастыратын негізгі (және екібағытты) желі.

Адрес желісі – жадыға (немесе сыртқы құрылғыларға) байланысымды қамтамасыз етуші желі.

Магистральдың байланыс жолдарының толық санын кеміту үшін көбіне адрес желісі мен дерек желісін кезектестіру тәсілі қолданылады, яғни байланыс жолдары арқылы әртүрлі уақыт мезетінде адрес пен дерек кезектеп жіберіледі (цикл басында – адрес, цикл аяғында – деректер).

Басқару желісінің жолдарындағы сигналдар кезекті цикл түрін анықтайды және оның әртүрлі бөліктеріне немесе кезеңдеріне сәйкесті уақыт мезеттерін бекітеді. Онымен қатар, басқару сигналдары процессор (немесе магистральдың өзге қожасының, бастаушының, master) жұмысын жадының немесе енгізу/шығару құрылғыларының (орындаушы құрылғының, slave) жұмысымен келістіреді.

Негізгі басқару сигналдары – алмастыру стробтары, яғни жазу (шығару) стробы мен оқу (енгізу) стробы болып табылады. Бұл сигналдарды процессор қалыптастырады, олар деректердің желі арқылы сәйкесті бағытта жіберілу мезеттерін анықтайды.

Енгізу/шығару құрылғыларын немесе сыртқы құрылғыларды (ЕШҚ немесе СҚ) жүйе магистралына қосу үшін олардың сигналдары сәйкесті енгізу/шығару интерфейстері арқылы жүзеге асырылатын, нақтылы стандарттарға сәйкес болуы керек.

Процессордың деректерді жазу және оқу операцияларын орындауы бағдарламалы-басқарылымды алмасу, үзіліс және жадыға тура шығу режимдерінде жүзеге асырылуы мүмкін, олардың ішіндегі ең бастысы – біріншісі.

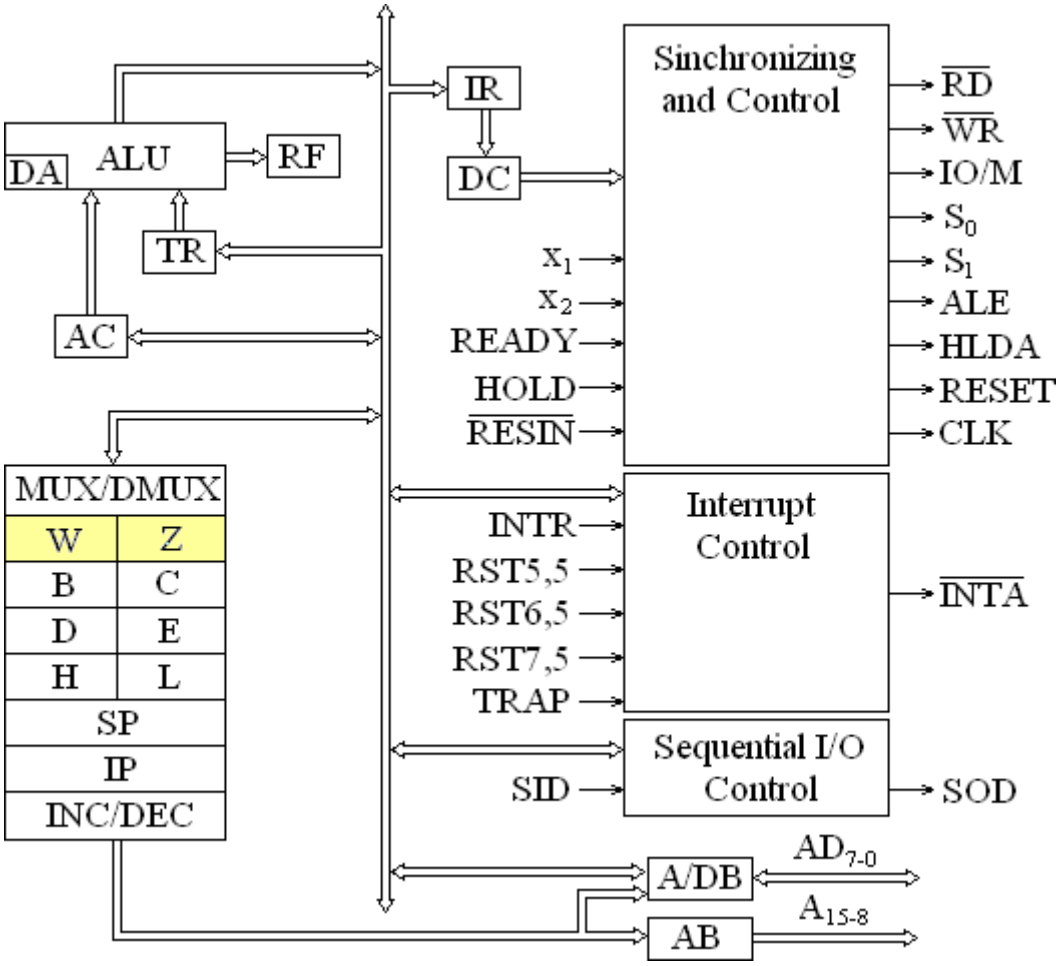
4.2 Intel 8085 микропроцессоры

Микропроцессор (МП) – күрделі цифрлық жүйелердің (мысалы, компьютерлердің) құрамындағы, ақпарат өңдеуші, басты блогы.

Микропроцессорлардың негізгі құрылым принциптері мен жұмыс тәртібін түсіну үшін олардың нақтылы бір түрін қарастыру ыңғайлы. Осындай үлгі ретінде қарапайым Intel 8085 микропроцессоры алынды. Өзінің қартаң жасына қарамай, бұл микропроцессор осы күнге дейін өз маңыздылығын жоймай, әлі де өндірісте шығарылады. Әрине, оның қолданылым аймағы – компьютерлер емес, оларда қазіргі заманда одан қуаттырақ және өнімділеу микропроцессорлар қолданылады. Intel 8085 сияқты микропроцессорлар, негізінде, олардың мүмкіншілігі жететін, басқару жүйелерінде пайдаланылады.

4.2.1 Микропроцессордың құрылымы.

Intel 8085 микропроцессорының құрылымы 4.1-суретте көрсетілген. Микропроцессордың құрама блоктарының арасындағы ақпарат алмастырылуы оның ішкі сегізразрядты дерек желісі арқылы жүзеге асырылады.



4.1 сурет

Микропроцессордың құрама блоктарын қарастыралық.

AC (Accumulator) – операндтардың біреуін және операция нәтижесін сақтаушы регистр-аккумулятор.

TR (Temporary Register) – операндтардың біреуін уақытша сақтауға арналған регистр.

ALU (Arithmetic-Logic Unit) – кірістеріне берілген екі сөзге (операндтарға) жүргізілетін операциялардың орындалуын жүзеге асыратын арифметикалық-логикалық құрылғы. Операцияға қатысты операндтардың біреуі уақытша регистрде (TR), ал екіншісі аккумуляторда сақталады; операция нәтижесі де аккумуляторға түседі. ALU тікелей қосу, алу, ығыстыру, сөздерді салыстыру операцияларын және логикалық операцияларды (инверсия, конъюнкция, дизъюнкция, екілік қосу) ғана орындайды. Олардан күрделірек операциялар (көбейту, бөлу және т.б.) қосалқы бағдарламалар арқылы орындалады. ALU құрамында екілік сандарды екілік-ондық кодқа айналдыру сұлбасы (DA, Decimal Adjust) болады.

RF (Register Flags) – орындалған арифметикалық және логикалық операциялардың нәтижелерінің белгі биттерін көрсететін, ту регистрі. Intel 8085 микропроцессорында бес түрлі белгі пайдаланылады: Z (Zero) – нөлдік белгісі, C (Carry) – тасымал белгісі, AC (Auxiliary Carry) – қосалқы тасымал белгісі, S (Sign) – терістік белгісі, P (Parity) – жұптық белгісі. Нөлдік белгі операция нәтижесінің нөл болғандығын, тасымал белгісі ең үлкен разрядтан шығарылатын тасымалды (яғни жетінші разрядтан; разряд нөмірінің нөлден басталатынын еске түсіріңіз), қосалқы тасымал белгісі үшінші разрядтан шығарылатын (яғни сегізразрядты сөздің тетрадаларының арасындағы) тасымалды (ол екілік-ондық кодтарға операциялар жүргізуге қажет), терістік белгісі – нәтиже таңбасын, жұптық белгісі нәтижедегі бірлік санының жұптығын сипаттайды. Бұл белгілер ақпараттың өңделу процесін басқару үшін пайдаланылады.

B, C, D, E, H, L – сегізразрядты жалпы қызмет регистрлері (ЖҚР), олар мультиплексор/демультиплексор (MUX/DMUX) арқылы ішкі дерек желісіне байланысқан. Олардың әрбіреуін жеке түрінде де, B-C, D-E, H-L түрінде қос-қостап (олар B, D, H жұптары деп аталады), 16-разрядты регистр ретінде де пайдалануға болады. H-L жұбы, әдетте, регистрлік жанама сілтеуге қажетті адресті сақтауға пайдаланылады.

W және Z – уақытша регистрлер, олар жадыдан команданың шығарылуы кезінде деректерді уақытша сақтауға арналған.

SP (Stack Pointer)–16-разрядты стек көрсеткіші. Стек сөз жинағын сілтеусіз сақтауға ыңғайланған жады құрылымы. Стектің негізгі қызметі – үзіліс бағдарламалары мен қосалқы бағдарламалардың жұмысын қамтамасыз ету. Стекпен жүргізілетін операциялар – Push (сөзді жазу) және Pop (сөзді оқу). Стек ретінде қызмет жадысының нақтылы аймағы бөлінеді де, оның түбі стек көрсеткішінде (SP) стек түбі ретінде белгіленеді. Стекпен жұмыс кезінде стек көрсеткішінде соңғы жазылған ұяшықтың адресі (стектің төбесі) тұрады. Push және Pop командаларының орындалуы кезінде SP мәні кеміп немесе өсіп

отырады. Жадының байтты ұйымдастырылымында және стекке регистр жұбындағы дерек енгізілу кезінде үлкен байт SP-1 адресі бойынша, ал кіші байт SP-2 адресі бойынша сақталады да, нәтижесінде SP мәні екіге кемиді. Оқу кезінде стектің жоғарғы екі ұяшығындағы дерек сәйкесті регистрлерге орналастырылады да, SP мәні екіге ұлғаяды. Сонымен, стек жұмысы LIFO (Last In – First Out) тәртібімен жүргізіледі, яғни стекке соңғы жазылған сөз бірінші шығарылады.

IP (Instruction Pointer)–16-разрядты команда санауышы. Онда кезекті орындалатын команданың адресі тұрады. Бастапқы қойылым кезінде ол нөл мәнін қабылдайды, демек бастапқы қойылым бағдарламасының бірінші командасы нолінші ұяшықта болады. Команда ұзындығы 1...3 байт болады. Жадыдан кезекті байт шығарылғаннан кейін PC мәні ұлғайтылады, команда бірбайтты болса, онда келесі команданың адресі, ал өзгеше жағдайда кезекті команданың келесі байтының адресі шығады. Команданың екінші және үшінші байттары W және Z регистрлеріне түседі.

INC/DEC (Increment/Decrement) сұлбасы ол арқылы жіберілген сөздерді бірге ұлғайтып немесе кемітіп отырады.

IR (Instruction Register) – команда регистрі. Ол жадыдан команданың бірінші байтын, яғни команда кодын қабылдайды.

DC – команда дешифраторы. Ол команда регистріндегі (IR) команда кодына сәйкесті операцияның машина циклдерін жүзеге асыруға қажетті сигналдар тудырады.

Synchronising and Control блогыкоманда дешифраторы мен машина циклдерінің шифраторының жағдайы негізінде ахуал сигналдары мен микропроцессорлық жүйені сыртқы құрылғылармен байланыстыратын сыртқы желіні басқару сигналдарын тудырады.

Interrupt Control және Sequential I/O Control– үзілісті басқару және тізбекті енгізілім/шығарылымды басқару блоктары. Олардың атқарар қызметі аттарынан-ақ түсінікті.

МП мен жады немесе СҚ арасындағы ақпарат алмасуы кезінде жадының сәйкесті ұяшығының немесе СҚ командасының құрамындағы немесе регистр жұптарының біреуіндегі адресі адрес буферіне (AB) жіберіледі. Үш жағдайлы адрес буфері адрестің үлкен разрядтарын адрес желісінің A_{15-8} жолдарына жібереді. Үш жағдайлы адрес/дерек буфері (A/DB) уақыт бойынша кезектесіп, адрестің кіші байтын немесе дерек байтын жібереді.

МП командаларының қалыпты жүргізілімінде олар жадыдан жазылым бағытымен HLT командасына жеткенше бір-бірден шығарылып орындалады. Жеке команданың шығарылуы мен орындалуы команда циклін (КЦ) құрады. Команда циклі бір немесе бірнеше машина циклінен (МЦ) тұрады. МП мен жады немесе СҚ арасында байт жіберуге арналған әрбір байланыс машина циклін қажет қылады. Өз кезегінде машина циклі бірнеше тактіге бөлінеді және такт саны машина циклінің түріне тәуелді болады.

Intel 8085 микропроцессорыныңкелесі түрдегі машина циклдері бар:

1) Команда таңдалымы (OF, Opcode Fetch).

2) Жадыдан оқу (MR, Memory Read).

3) Жадыға жазу (MW, Memory Write).

4) Сыртқы құрылғыдан оқу (IOR, Input-Output Read).

5) Сыртқы құрылғыға жазу (IOW, Input-Output Write).

6) Үзілісті құптау (INA, Interrupt Acknowledge).

7) Желіні босату (BI, Bus Idle).

8) Тоқтатылым (HALT).

Әрбір машина циклінің басында цикл түрін сипаттаушы ахуал сигналдары тудырылады. Олар сыртқы құрылғыға жіберілетін қосалқы басқару сигналдарын қалыптастыруда пайдаланылады.

4.2.2 Микропроцессордың шықпалары мен сигналдары.

A_{15-8} – жады адресінің үлкен байтын немесе сыртқы құрылғының толық адресін шығаруға арналған үш жағдайлы шығыс жолдары. HOLD, HALT және RESET режимдерінде олардың шықпалары үшінші (жоғары кедергілі) жағдайға ауысады;

AD_{7-0} – машина циклінің бірінші тактісінде жады адресінің кіші байтын немесе сыртқы құрылғының толық адресін шығаруға арналған үш жағдайлы кезектелімді жолдар. Адрес жіберілу тактісінен кейін олар дерек желісі ретінде пайдаланылады;

ALE – машина циклінің бірінші тактісінде жіберілген жады адресінің кіші байтын сыртқы адрес регистріне енгізуге рұқсат беру стробы. Адресті регистрге енгізу ALE сигналының артқы ауытқымасымен жүзеге асырылады;

$\overline{RD}$, $\overline{WR}$ – сілтенген жады ұяшығына немесе сыртқы құрылғыға жіберілетін оқу және жазу стробтары. HOLD, HALT және RESET режимдерінде олардың шықпалары үшінші (жоғары кедергілі) жағдайға ауысады;

READY – жадының немесе сыртқы құрылғының МП-мен ақпарат алмасуға дайын екендігін көрсететін кіріс сигналы. Жадының немесе сыртқы құрылғының дайын еместігінде, МП күту жағдайына кіреді де, бұл жағдай бірлік деңгейдегі READY сигналы түскенше созылады;

S_1, S_0 – МП-дың сыртқы құрылғыға жіберетін ахуал сигналдары. Олар машина циклінің басында қалыптасып, бұл цикл біткенше сақталады;

IO/M – жадыны немесе сыртқы құрылғыны таңдау сигналы. Оның жоғарғы деңгейінде сыртқы құрылғыға, ал төменгі деңгейінде жадыға байланыс жүзеге асырылады.

S_1, S_0 сигналдарымен бірге IO/M сигналы машина циклінің түрін анықтайды. Ахуал сигналдары мен $\overline{RD}$, $\overline{WR}$ және $\overline{INTA}$ сигналдарының әртүрлі машина цикліне байланысты мәндері 4.1 кестеде келтірілген.

4.1 к е с т е

МЦ	Ахуал сигналдары			Басқару сигналдары		
	IO/M	S_1	S_0	$\overline{RD}$	$\overline{WR}$	$\overline{INTA}$
OF	0	1	1	0	1	1
MR	0	1	0	0	1	1
MW	0	0	1	1	0	1
IOR	1	1	0	0	1	1
IOW	1	0	1	1	0	1
INA	1	1	1	1	1	0
BI	3S	x	x	1	1	1
HALT	3S	0	0	3S	3S	1

Кестеде 3S арқылы үшінші (жоғары кедергілі) жағдай белгіленген.

x_1, x_2 – микропроцессордың ішкі синхроимпульстер генераторына қосылатын жұмысын қамтамасыз етуші кварцты резонатордың немесе одан басқа жиілік тиянақтаушы тізбектердің қосылатын шықпалары;

$\overline{RESIN}$ ($\overline{RESET\ IN}$) – микропроцессорды бастапқы жағдайына қайтару кірісі. Бұл сигнал қорек көзі қосылған кезде өздігінен қалыптасады, оның кез келген уақытта оператордың командасы арқылы да түсуі мүмкін. Оның әрекетімен PC және IR регистрлері, үзіліс рұқсаты мен желі ұсталымының құпталым триггерлері және т.б. тазартылады;

CLK – микропроцессорлық жүйеге жіберілетін синхроимпульстер шығысы. Бұл импульстердің жиілігі микропроцессордың x_1 және x_2 шықпаларындағы жиіліктен екі есе кем болады;

RESET – жүйенің сыртқы модульдерін тазарту сигналы, ол тактілік CLK импульстеріне байланыстырылған және оның фазасы $\overline{RESIN}$ сигналынан өзгеше болады;

INTR (Interrupt Request) – векторлық үзіліс сұранысының кірісі. Бағдарлама арқылы үзіліске рұқсат берілген жағдайда, бұл сигнал $\overline{INTA}$ стробын тудырады. Бұл сұранысқа сәйкес шақырылатын қосалқы бағдарламаның адресін сыртқы құрылғы жібереді. Тазартылым кезінде сұраныс сигналы қабылданбайды (үзіліске тыйым салынған);

$\overline{INTA}$ (Interrupt Acknowledge) – кезекті команда циклі біткеннен кейін жіберілетін векторлық үзілісті құптау стробы. Ол үзіліс векторын оқу үшін пайдаланылады;

RST 5,5; RST 6,5; RST 7,5 – RSTn (n = 5,5; 6,5; 7,5) түріндегі радиалды үзіліс сұранысының кірістері. Бұл сұраныстарға сәйкесті қызмет атқаратын қосалқы бағдарламалардың бастапқы адресі – 8n. Кіріс мәртебелері бекітілген, ең жоғарғы мәртебе RST 7,5 кірісінде. Бұл сұраныс тобының мәртебесі INTR сұранысының мәртебесінен жоғары болады. Бұл сұраныстардың әрбіреуіне және бір-біріне тәуелсіз тыйым салуға болады;

TRAP – мәртебесі ең жоғарғы дәрежедегі, тыйым салынбайтын үзіліс сұранысының кірісі;

SID, SOD (Serial Input Data, Serial Output Data) – деректерді тізбекті жіберу кірісі мен шығысы. RIM командасымен кіріс биті аккумулятордың жоғарғы разрядына енгізіледі, SIM командасымен шығыс биті осы разрядтан шығарылады;

HOLD – желіні иелену сұранысы. Оны сыртқы құрылғы қалыптастырады;

HLDA (Hold Acknowledge) – желінің иеленілуін құптау сигналы. Ол HOLD сигналына жауап ретінде кезекті машина циклінің аяғында қалыптастырылады да, микропроцессордың жүйе желісінен ажыратылғандығын құптайды. Бұл кезде желілер мен басқарушы $\overline{RD}$, $\overline{WR}$, IO/M және ALE сигналдарының жолдары үшінші жағдайға ауыстырылады.

4.2.3 Микропроцессор жұмысының ұйымдастырылуы.

Команда циклі (КЦ) команда таңдалымынан басталады (Opcode Fetch, OF). Бірінші машина циклі Ml әрқашан OF, онда МП команданың бірінші байтын алады. Одан кейін, команданың бірбайтты, екібайтты немесе үшбайтты болуына байланысты бір немесе екі MR (Memory Read) түрлі машина циклінің болуы мүмкін.

Команданың бірінші байтында (b_1) операция коды, сілтеу тәсілі жөніндегі мәліметтер, ал егер команда бірбайтты болса, онда операндтың адресі де болады. Бұндағы адрес қысқа адресті “регистр-регистр” түріндегі операцияларға ғана байланысты болады. Жалпы қызметтік 8 регистрді сілтеуге – үшразрядты, ал регистр жұптарын сілтеуге екіразрядты адрес жеткілікті. Команданың екінші байтында (b_2) операндқа қатысты жады ұяшығының адресінің кіші бөлігі немесе сыртқы құрылғының адресі немесе операндтың тікелей өзі болады (соңғылар команданың екібайттылық жағдайында). Команданың үшінші байтында (b_3) операнд адресінің үлкен бөлігі немесе тікелей берілетін деректің екінші байты (регистрлер жұбына енгізу кезінде) болады. Регистрлер мен регистр жұптарының адрестері 4.2 кестеде берілген.

4.2 к е с т е

Регистрлер								Регистр жұптары			
B	C	D	E	H	L	M	A	B	D	H	SP
000	001	010	011	100	101	110	111	00	01	10	11

Команданың таңдалып, айқындалғанынан кейін, оны орындауға қосымша машина циклдерінің қажет болуы мүмкін. Командада небәрі бірден беске дейін машина циклі болуы мүмкін.

Машина циклі төменде қарастырылатын қалыпты әрекеттер жүргізілетін тактілерден тұрады. Әртүрлі машина цикліндегі такт саны – 3…6, бірақ олардың көпшілігі үш тактіден тұрады.

Команда циклінде 4...18 такт болуы мүмкін.

Әртүрлі машина циклін жүзеге асырушы сигналдарды команданың бірінші байтындағы ақпарат негізінде МП тудырады.

Айтылғанды мысал арқылы суреттелік. Аккумулятордағы деректі жады ұяшығына тура сілтеумен (яғни адрестің командада көрсетілуі арқылы) жіберуші STA D3D2 (Store Accumulator Direct) командасының орындалуын қарастыралық. Команда – үшбайтты, оның МП-ға жіберілуі үшін үш машина циклі керек, оның біріншісінде b_1 байты команда регистріне (IR) жіберіліп, келесілерінде b_2 және b_3 байттары уақытша регистрлерге (W және Z) жіберіледі. Бүкіл команда қабылданғаннан кейін МП аккумулятордағы дерек байтын, адресі МП-ға түскен, жады ұяшығына жіберу жұмысын жүзеге асырады. Сонымен, команда циклі OF-MR-MR-MW тәртібімен жүргізілетін төрт машина циклінен құрылады.

Әрбір машина циклі тактілік импульстердің аттас ауытқымаларының аралығымен анықталатын тактілерге (жағдайларға) бөлінеді.

Машина циклінің тактілерінде орындалатын қалыпты әрекеттер:

T_1 – жадының немесе СҚ-ның адресі AD_{7-0} және A_{15-8} желілеріне қойылып, AD_{7-0} биттерін бекітуге ALE сигналы тудырылады. IO/M, S_1 және S_0 жолдарында цикл түрін анықтаушы ақпарат қойылады. HALT туы тексеріледі.

T_2 – Ready және Hold кірістері тексеріледі. Егер осы машина циклі команда таңдалымының бір бөлігі болса, онда бағдарлама санауышы бірге ұлғайтылады. BI (желінің босатылуы) циклінен басқа машина циклдерінде басқару стробтарының ($\overline{RD}$, $\overline{WR}$ немесе $\overline{INTA}$) біреуі бірлік жағдайынан жандандырылған нөлдік жағдайына ауысады.

T_W (Wait) – жадының немесе СҚ-ның дерек алмасуға дайын еместігінде (READY жолында төменгі деңгейлі кернеу тұрғанда) туады. Адрес, дерек және басқару жолдарының жағдайы такт аяғындағы қалпында қалады. READY сигналы әрбір күту тактісінде тексеріледі.

T_3 – команда немесе дерек байты микропроцессорға немесе одан жіберіледі. Жандандырылған басқару стробының деңгейі бірлік деңгейіне өзгертіледі.

T_4 – команда регистріндегі код айқындалады.

$T_{5,6}$ – кейбір командалардың бітірілуіне қажет кезінде пайдаланылады.

Жүйе желілері пайдаланылмайды.

Машина циклінде әрқашан T1....T3 тактілері болады, кейде такт саны одан көбірек болуы мүмкін, бірақ оқу мен жазуға үш такт ғана қажет.

4.2.4 Үзіліс жүйесі.

Микропроцессорлық жүйенің жұмысы кезінде оның ішінде немесе сыртында кезексіз әрекет қажетті оқиғалар тууы мүмкін. Бұндай әрекет негізгі бағдарламаны үзіп, үзіліс сұранысының жұмысын орындауға ауысу арқылы жүзеге асырылады. Жүйе ішіндегі сұраныстар оның жұмыс ахауынан, разряд торының асып кетуінде, нөлге бөлу кезінде және сыртқы құрылғылардың қызмет талабынан туады. Сыртқы сұраныстардың басқарылым объектілерінің қауіптілік жағдайында қорек көздерінің ахаулығында және т.б. жағдайларда түсуі мүмкін.

Баяу істейтін сыртқы құрылғылардың сұраныстары бойынша үзілістер, микропроцессор уақытын олардың ақпарат алмасуға дайын кезінде ғана алуына мүмкіндік беріп, бұндай ұйымдастырылым нәтижесінде жалпы жүйенің жұмыс тиімділігі көтеріледі. Сыртқы құрылғыға микропроцессор қажет болғанда, ол үзіліс сұранысының триггерін бірлік жағдайға қояды да, бұл үзіліс сигналын микропроцессордың қабылдап, сәйкесті өңдегенінше ол осы жағдайында қалады. Қабылданған үзіліс сұранысына жауап ретінде микропроцессорлық жүйеде кезекті команданың орындалуы бітіріледі, микропроцессор жағдайы еске сақталады, үзіліс қызметінің қосалқы бағдарламасы орындалады, сосын жүйенің басқарылуы негізгі бағдарламаның сәйкесті командасына қайтарылады.

Intel 8085 микропроцессорының бес үзіліс кірісі мен оларды басқарушы INTA шығысы болады. Үзіліс, оған рұқсаттың бар кезінде, CALL командасының әрекетімен басталады: бағдарлама санауышының (PC) жағдайы стекке жіберіледі, ал оның өзіне орындалатын қосалқы бағдарламаның адресі енгізіледі де, үзіліс сұранысына қажетті жұмыс орындалады.

Микропроцессордың үзілістерге байланысты кірістері: TRAP; RST 5,5; RST 6,5; RST 7,5; INTR. Үзіліс жүйесінің ұйымдастырылымында сұраныстарды жасыру мәселелері шешіліп, олардың мәртебелік деңгейлері анықталады.

Жасыру дегеніміз – сәйкесті кірістің әрекетіне тыйым салу. Үзіліс сұраныстарының кірістерінің жасырылымды немесе жасырылмайтын (яғни әрқашан қабылдайтын жағдайында) болуы мүмкін.

TRAP кірісі – жасырылмайтын кіріс және оның мәртебесі ең жоғарғы деңгейде болады. Оған бағдарлама командалары арқылы тыйым салынбайды. Бұл кіріске микропроцессорлық жүйедегі, кезексіз әрекет қажет қылатын маңызды оқиғалардың (мысалы, қорек көзінің ақауының) хабарлау сигналдары беріледі.

TRAP үзілісіне сайланысты қызметтің қосалқы бағдарламасының бастапқы адресі жадының бекітілген, адресі 24H ұяшықта орналастырылған.

RSTn (n = 5,5; 6,5; 7,5) кірістерінің белгілемесі Restart сөзінен алынған. Бұл кірістердің үзілістері – жасырылмалы, яғни оларға EI (Enable Interrupt) және DI (Disable Interrupt) командалары арқылы рұқсат беруге немесе тыйым салуға болады (олардың бәріне қатарынан). Микропроцессордың бастапқы тазартылымы бұл сұраныстардың барлығына да тыйым салады, оларға бұдан әрі рұқсат беру үшін EI командасын беру керек. RSTn сұраныстарына жеке тыйым салу мүмкіндігі де бар, ол арнайы SIM (Set Interrupt Mask) командасымен аккумулятордың A0...A2 биттерінің алдынала енгізілген мәндеріне сәйкесті қойылады. RSTn кірістерінің мәртебелері RST 7,5; RST 6,5; RST 5,5 тәртібінде бекітілген. RSTn үзілістерінің қызметінің қосалқы бағдарламаларының бастапқы адрестері n мәнімен анықталады, сәйкесті бағдарламаның жұмысы бағдарлама санауышына 8n мәнін (002CH, 0034H немесе OO3CH) енгізуден басталады. Өзгеше айтқанда, бұл кірістердің үзіліс

векторлары өздігінен анықталады, яғни оларды сыртқы құрылғыдан жіберудің қажеті болмайды.

RST 7,5 кірісі – динамикалық кіріс, ол сигналдың тура ауытқуына әрекет етеді, яғни сұраныс сигналы динамикалы кірісті триггерге енгізіледі де, оның алып тасталғанында, бұл үзіліс сұранысының қызметі орындалғанша немесе SIM немесе RESET командаларына дейін онда сақталады. RST 6,5 және RST 5,5 кірістері – статикалық кірістер, олар сигнал деңгейіне әрекет етеді, демек сұраныс сигналы жойылысымен, алып тасталынады.

INTR (Interrupt) кірісіне түсетін сұраныстың үзіліс векторы мкропроцессорға сырттан жіберілу керек. Бұл кіріске көптеген жағдайда үзілістер контроллері қосылады, ол бірнеше сыртқы құрылғылардың сұраныстарын қабылдап, олардың мәртебелік және жасырым мәселелерін шешіп, мкропроцессорға жіберілетін жалғыз INTR сигналы мен сұранысқа сәйкесті үзіліс векторын қалыптастырады. Бұл жағдайда да RSTn командасы орындалады, бірақ n мәні үзіліс көзіне тәуелді болады. Үзілістерге рұқсат бар кезінде INTR сұранысының түсуіне жауап ретінде, микропроцессор $\overline{\text{INTA}}$ сигналымен үзіліс берілгендігін хабарлайды. Бұл сигнал буферлік күшейткіштердің шығыстық рұқсат кірістеріне түседі.

Үзілістің өңделуі кезінде, EI командасы орындалғанша, TRAP үзілісінен басқа үзілістерге тыйым салынады. Жасырылмайтын TRAP үзілісі өзге үзілістерді жібергізбейді, бірақ бұрынғы түскен сұраныстың рұқсаттық жағдайын сақтайды.

4.2.5 Тізбекті енгізу/шығару жүйесі.

Микропроцессордың тізбекті деректерді жіберуге арналған екі кірісі бар: SOD және SID (Serial Output Data және Serial Input Data).

SOD шықпасы SIM командасымен басқарылады, ал SID шықпасындағы сигнал RIM командасымен оқылады. Бұл командалар бұрын RSTn үзілістерінің жасырылымын қою және тазарту командалары ретінде таныстырылғанды, олар тізбекті енгізу/шығаруды басқаруға да пайдаланылады.

SIM командасының орындалуынан бұрын аккумуляторда басқару сөзі қалыптастырылады (4.3 кесте).

4.3 к е с т е

7	6	5	4	3	2	1	0
SOD	SOE	x	R7,5	MSE	M7,5	M6,5	M5,5

Басқару сөзінің биттері келесі мағынада түсініледі:

а) SOD – деректердің тізбекті шығысы;

б) SOE – бірлік мәнінде тізбекті SOD деректерін микропроцессордың сәйкесті шығысына жіберетін сигнал;

в) R7,5 сәйкесті RST7,5 кірісін тазартады (осы кірістегі сигналдың динамикалы басқарылымды триггерге қабылданатындығын еске түсіреміз);

г) MSE (Mask Set Enable) – 2...0 биттерінің әрекетін жандандырушы сигнал;

д) M7,5...M5,5 биттерінің бірлік мәні оларға сәйкесті RST7,5...RST5,5 сұраныстарын жасырады, яғни оларға тыйым салады.

Мысалы, SOD = 1, RST6,5 кірісіне рұқсат берілуі, RST7,5 триггерінің тазартылуы және RST7,5 мен RST5,5 кірістерінің жасырылуы екі командамен жүзеге асырылады:

```
MVI A, b2        ; аккумулятор биттерін қою
SIM              ; жасырым биттерін және SOD битін өзгерту
```

MVI A, b_2 командасы аккумуляторға b_2 байтын жібереді, яғниаккумуляторға тіке сілтеу арқылы дерек жіберу операциясын (A) ← (b_2) орындайды. Бұл жағдайдағы b_2 байтының түрі: 11x11101.

Тізбекті деректерді SID шықпасы арқылы енгізу үшін RIM командасы пайдаланылады. RIM командасының орындалғанынан кейін аккумуляторда биттерінің мағынасы 4.4 кестеде келтірілген сөз қалыптасады:

а) SID – тізбекті енгізілетін деректер;

б) I7,5; I6,5; I5,5 – RST7.5, RST6.5 және RST5.5 шықпаларындағы логикалық деңгейлер;

в) IE – үзіліске рұқсат сигналы;

г) M7,5...M 5,5 – маскалардың логикалық деңгейлері.

4.4 к е с т е

7	6	5	4	3	2	1	0
SID	I7,5	I6,5	I5,5	IE	M7,5	M6,5	M5,5

I7,5... I5,5 биттері сәйкесті логикалық деңгейлерді RIM командасының кезінде қалыптастырады. IE биті EI және DI командаларының қайсысы соңғы орындалғандығын көрсетеді, оған бұл кезде үзіліс режимінің болуы да (оның IE триггерін тазартып, басқа үзілістерге тыйым салатындығына байланысты) әсер етеді. M7.5...M5,5 биттері үзіліс маскаларының кезекті жағдайларын суреттейді.

4.3 Intel 8085 микропроцессорының командалар жүйесі

МП командалары 4.5 кестеде келтірілген. Кестенің бірінші бағанасында командалардың мнемоникасы (жазылым түрі) берілген. Онда регистрлер – r, регистр жұптары – r_p, жанама сілтеумен байланысатын жады ұяшығы – M, команданың үшінші және екінші байттары – b_3b_2, сыртқы құрылғының адресі – port арқылы берілген. M белгісімен жады ұяшығына жанама сілтеу көрсетіледі, бұл жағдайда жады ұяшығының адресі H регистрлік жұбынан алынады да, демек командада адрес көрсетілуі қажет емес.

4.5 к е с т е

Мнемокод	Код	Тулар	Байт	Такт	Цикл	Түсініктеме
1	2	3	4	5	6	7
Дерек тарату командалары						
MOV r1, r2	01DDDSSS	–	1	4	1	$r_1 \leftarrow (r_2)$
MOV M, r	01110SSS	–	1	7	2	[(HL)] ← (r)
MOV r, M	01DDD110	–	1	7	2	(r) ← [(HL)]
MVI r, b2	00DDD110	–	2	7	2	r1 ← b2
MVI M, b2	36	–	2	10	3	[(HL)] ← b2
LXI rpb3b2	00PP0001	–	3	10	3	rp ← b3b2
LDA b3b2	3A	–	3	13	4	A ← b3b2
STA b3b2	32	–	3	13	4	b3b2 ← (A)
LHLD b3b2	2A	–	3	16	5	HL ← b3b2
SHLD b3b2	22	–	3	16	5	b3b2 ← (HL)
LDAX rp	00PP0010	–	1	7	2	A ← [(rp)]
STAX rp	00PP1010	–	1	7	2	[(rp)] ← (A)
XCHG	EB	–	1	4	1	(HL) ↔ (DE)
Арифметикалық және логикалық операциялардың командалары						
ADD r	10000SSS	+	1	4	1	A ← (A) + (r)
ADD M	86	+	1	7	2	A ← (A) + [(HL)]
ADI b2	C6	+	2	7	2	$A \leftarrow (A) + b_2$
ADC r	10001SSS	+	1	4	1	$A \leftarrow (A) + (r) + (T_C)$
ADC M	8E	+	1	7	2	$A \leftarrow (A) + [(HL)] + (T_C)$
ACI b2	CE	+	2	7	2	$A \leftarrow (A) + b_2 + (T_C)$
SUB r	10010SSS	+	1	4	1	A ← (A) – (r)
SUB M	96	+	1	7	2	A ← (A) – [(HL)]
SUI b2	D6	+	1	7	2	A ← (A) – b2
SBB r	10011SSS	+	1	4	1	$A \leftarrow (A) – (r) – (T_C)$
SBB M	9E	+	1	7	2	A ← (A) – [(HL)] – (TC)
SBI b2	DE	+	2	7	2	A ← (A) – b2 – (TC)
INR r	00DDD100	(+)	1	4	1	r ← (r) + 1
INR M	34	(+)	1	10	3	[(HL)] ← [(HL)] + 1
DCR r	00DDD101	(+)	1	4	1	r ← (r) – 1
DCR M	35	(+)	1	10	3	[(HL)] ← [(HL)] – 1
INX rp	00PP0011	–	1	6	1	rp ← (rp) + 1
DCX rp	00PP1011	–	1	6	1	rp ← (rp) – 1
DAD rp	00PP1001	–	1	10	3	[(HL)] ← [(HL)] + (rp)
DAA	27	+	1	4	1	A ← (A)2-10
ANA r	10100SSS	+	1	4	1	A ← (A) ∧(r)
ANA M	A6	+	1	4	1	A ← (A) ∧[(HL)]
ANI b2	E6	+	2	7	2	A ← (A) ∧b2
XRA r	10101SSS	+	1	4	1	A ← (A) $\oplus$ (r)
XRA M	AE	+	1	7	2	A ← (A) $\oplus$ [(HL)]
XRI b2	EE	+	2	7	2	A ← (A) $\oplus$ b2
ORA r	10110SSS	+	1	7	2	A ← (A) ∨(r)

4.5 к е с т е (жалғасы)

1	2	3	4	5	6	7
ORA M	B6	+	1	7	2	A ← (A) ∨[(HL)]
ORI b2	F6	+	2	7	2	A ← (A) ∨b2
CMP r	10111SSS	+	1	4	1	(A) – (r)
CMP M	BE	+	1	4	1	(A) – [(HL)]
CPI b2	FE	+	2	7	2	(A) – b2
CMA	2F	–	1	4	1	$A \leftarrow \overline{(A)}$
STC	37	C	1	4	1	TC ← 1
CMC	3F	C	1	4	1	$TC \leftarrow \overline{(T_C)}$
RLC	07	C	1	4	1	A7-1 ← (A6-0), A0 ← (TC), TC←A7
RRC	0F	C	1	4	1	A6-0 ← (A7-1), A7 ← (TC), TC←A0
RAL	17	C	1	4	1	A7-1 ← (A6-0), A0 ← (A7), TC←A7
RAR	1F	C	1	4	1	A6-0 ← (A7-1), A7 ← (A0), TC←A0
Басқару командалары						
JMP b3b2	C3	–	3	10	3	PC ← b_3b_2
Jcond b3b2	11CCCC01	–	3	10	3	Cond = 1: PC ← b3b2
CALL b3b2	CD	–	3	18	5	SP ← (PC), PC ← b3b2
Ccond b3b2	11CCC100	–	3	2/5	9/18	Cond = 1: SP ← (PC), PC ← b3b2
RET	C9	–	3	10	3	PC ← [(SP)]
Rcond	11CCC100	–	3	17/11	5/3	Cond = 1: PC ← [(SP)]
RST n	111nnn111	–	1	11	3	PC ← 8n
SPHL	E9	–	1	6	1	SP ← (HL)
Арнайы командалар						
PUSH r_p	11PP0101	–	1	11	3	SP ← (SP) – 2; [(SP)]; [(SP) + 1]← (r_p)
PUSH PSW	F5	–	1	11	3	SP ← (SP) – 2; [(SP)], [(SP) + 1]←(A),(F)
POP rp	11PP0001	–	1	10	3	r_p ← [(SP)],[(SP) + 1], SP ← (SP) + 2
POP PSW	F1	–	1	10	3	F,A ← [(SP)], [(SP) + 1]; SP ← (SP) + 2
XTHL	E3	–	1	18	5	(HL) ↔ [(SP)], [(SP) + 1]
PCHL	F9	–	1	5	1	PC ← (HL)
IN port	DB	–	2	10	3	A ← (port)
OUT port	D3	–	2	10	3	port ← (A)
EI	FB	–	1	4	1	Үзіліске рұқсат беру
BI	F3	–	1	4	1	Үзіліске тыйым салу
HLT	76	–	1	7	2	Тоқтатылым

4.5 к е с т е (жалғасы)

1	2	3	4	5	6	7
NOP	00	–	1	4	1	Операция жоқ
RIM	20	–	1	4	1	Үзіліс маскасын оқу
SIM	30	–	1	4	1	Үзіліс маскасын жазу

Екінші бағанадағы команда кодтары операндтардың адрестің көрсетілуі қажет жағдайда, сегізразрядты екілік сан түрінде, өзге жағдайларда 16-лық сан түрінде берілген. Регистрлердің жалпылама адрестерінің разрядтары дерек көзі үшін S (Data Source) символдарымен, деректің жеткізілетін жері D (Data Destination) символдарымен, регистр жұптары P (Pair) символдарымен бейнеленген. Бұл символдардың орнына командағы қатысты регистрлер мен регистр жұптарының нақтылы адресін қою арқылы команданың коды алынады. Шартты ауыстыру командаларының кодындағы шарттардың коды C (Condition) символдарымен белгіленген, олардың түрлері мен белгілемелері 4.6 кестеде келтірілген.

4.6 к е с т е

CCC	Мнемокод (cond)	Шарттар
000	NZ	Нөлге тең болмау
001	Z	Нөлге тең болу
010	NC	Тасымалдың тумауы
011	C	Тасымалдың тууы
100	PO	Тақтық
101	PE	Жұптық
110	P	Плюс
111	M	Минус

Операцияның орындалу белгілері ту регистрінде қалыптасады (4.7 кесте).

4.7 к е с т е

7	6	5	4	3	2	1	0
S	Z	0	AC	0	P	1	C

RST командасының кодындағы n әріптерімен белгіленген үш разрядтың мәнін үзіліс жүйесі қояды немесе оны бағдарлаушы көрсетеді.

Салыстыру операцияларында операндтарға алу операциясы жүргізіледі де, оның нәтижесіне байланысты Z және S белгілері қойылады (бірақ нәтиже ешқайда сақталмайды).

Үшінші бағанадағы “–” символы команданың орындалуында ешқандай ту көтерілмейтіндігін, “+” символы барлық белгінің қойылатындығын, “(+)” символы тасымал белгісінен басқа белгілердің қойылатындығын, “C” символы тасымал белгісінің ғана қойылатындығын білдіреді.

4.4 Сілтеу тәсілдері

Бағдарламадағы нақтылы командаға сәйкесті операцияның орындалуы үшін, онда команданың түрін анықтаушы кодымен қатар, операцияға қатысты операндтардың қайдан алынатындығы және орындалған операцияның нәтижесінің қайда орналастырылатындығы көрсетілуі керек. Intel 8085 микропроцессорында сілтеу адрестерін қалыптастыру төрт түрлі тәсілмен жүзеге асырылады, олар – тура сілтеу, регистрлік сілтеу, жанама сілтеу және тіке сілтеу тәсілдері.

Тура сілтеу тәсілінде адрес коды орындалушы команданың құрамында болады. Үлкен көлемді жадының адрес разрядтарының саны да үлкен болатындықтан, ол команданы ұзартады. Мысалы, 2^{16} ұяшықтан тұратын жады сілтеу 16-разрядты адреспен сілтенеді. Сондықтан бұндай командалар үшбайтты болады. Сыртқы құрылғыларға сілтеу сегізразрядты адрес арқылы жүзеге асырылады да, бұндай командалар екібайтты болады.

Регистрлік сілтеу тәсілінде операнд микропроцессордың ішкі регистрлерінің біреуінде болады. Intel 8085 микропроцессорының ішкі регистрлерінің саны сегізден аспайтындықтан оларға сілтеуге үшразрядты адрес жеткілікті болады, сондықтан бұндай командалар бірбайтты болады.

Жанама сілтеу тәсілінде командада орындалатын операцияға қатысты операндтың адресі тұрған регистр көрсетіледі. Бұндай командалар да тура сілтемді командалар сияқты ықшамды келеді, бірақ олардың орындалуы үшін бұл регистрге (жанама сілтеу регистріне) қажетті адрес алдынала енгізілуі керек. Жанама сілтеу регистрі ретінде Intel 8085 микропроцессорында HL жұбы пайдаланылады. Жанама сілтеу тәсілі тізбе түрінде берілген деректерді өңдеуге аса ыңғайлы келеді, бұл кезде адрес бір рет қойылады да, кезекті адрес оның алдыңғы мәнін бірге өзгерту арқылы алынады.

Тіке сілтеу тәсілінде операцияға қатысты операнд тікелей команда құрамында беріледі. Бұндай командалар екібайтты (сыртқы құрылғымен байланысқа) немесе үшбайтты (жадымен байланысқа) болады.

Суреттелген сілтеу тәсілдерінен күрделірек келетін сілтеу тәсілдері де (индексті сілтеу, қатынасты сілтеу және т.б.) бар, бірақ олар Intel 8085 микропроцессоры сияқты қарапайым микропроцессорларда пайдаланылмайды.

Әртүрлі сілтеу тәсілдерін пайдалану бағдарламалардың көлемі мен олардың орындалу уақытын қысқартуға мүмкіндік береді.

5 Микропроцессорлық жүйелерді бағдарлау

Микропроцессорлық жүйенің жұмысы шешілетін мәселеге байланысты командалар тізбесі түрінде құрылған бағдарламаның басқаруымен жүзеге асырылады. Бұндай бағдарламалар, әдетте, командалардың мнемокодтары арқылы суреттелетін ассемблер тілінде жазылады. Мнемокодтарды машина тіліне (екілік кодтарға) аударушы (translation) арнайы бағдарлама да

Ассемблер деп аталады. Машина тіліндегі бағдарлама микропроцессорлық жүйенің жадылық модулінде сақталады да, оның құрамындағы команда кодтары нақтылы тәртіппен таңдалып (оқылып), ол команда кодтарында берілген мәліметтер (операция коды, сілтеу тәсілі, деректер) негізінде орындалады.

Ассемблер тіліндегі бағдарлама нақтылы тәртіппен жазылады.

{Метка:} Мнемокод {Операнд(тар)} ; { Түсініктеме }

Көрнекі жақшаларда болуы да, болмауы да мүмкін элементтер келтірілген. Белгі, мнемокод және операндтар бос орын арқылы ажыратылады.

Белгі – команданың символды адресі. Белгілер ауыстыру арқылы баратын командаларда ғана қойылады. Ауыстыру немесе қосалқы бағдарламаларды шақыру командаларында белгі аталымы операнд ретінде пайдаланылады.

Мнемокод – команданың ассемблер тіліндегі жазылымы. Мнемокод ретінде команда қызметінің ағылшын тіліндегі мағынасының қысқартылымы пайдаланылады: ADD (ADDition), SUB (SUBtraction), XCHG (eXCHanGe).

Операндтар үтір арқылы бөлінеді. Екі операнд берілген жағдайда оның біріншісі ақпарат қабылдаушысы, ал екіншісі ақпарат көзі болады.

Түсініктеме командалардың қызметін суреттеуге ғана пайдаланылады, ол машина тіліне аударылмайды және әрқашан үтір-нүктеден басталады.

Бұдан әрі бірнеше қарапайым есептерді шешуге арналған бағдарламалар құрамыз. Бұл бағдарламалардың кемістігін анықтап, олардың кейбір командаларын басқа командалармен ауыстыру арқылы жетілдіріп, микропроцессордың командалар жүйесіндегі әртүрлі командалардың қызметі мен қолданылым жағдайына байланысты ыңғайлылықтарын түсіндіреміз.

Құрылған бағдарламалардың жұмысқа жарамдылығын, яғни оларда орфографиялық және іскерлік қателіктердің жоқтығын сәйкесті бағдарламалы симулятор (8085 Simulator IDE) арқылы тексеруге болады.

Симулятордың құрылымы, онымен жұмыс істеу тәртібі және оның іскерлік мүмкіндіктері (оның құрамындағы әртүрлі аспап-құралдармен жұмыс істеу) сәйкесті бағдарламаларды талқылау барысында қарастырылады.

5.1 Дерек таратылымдары

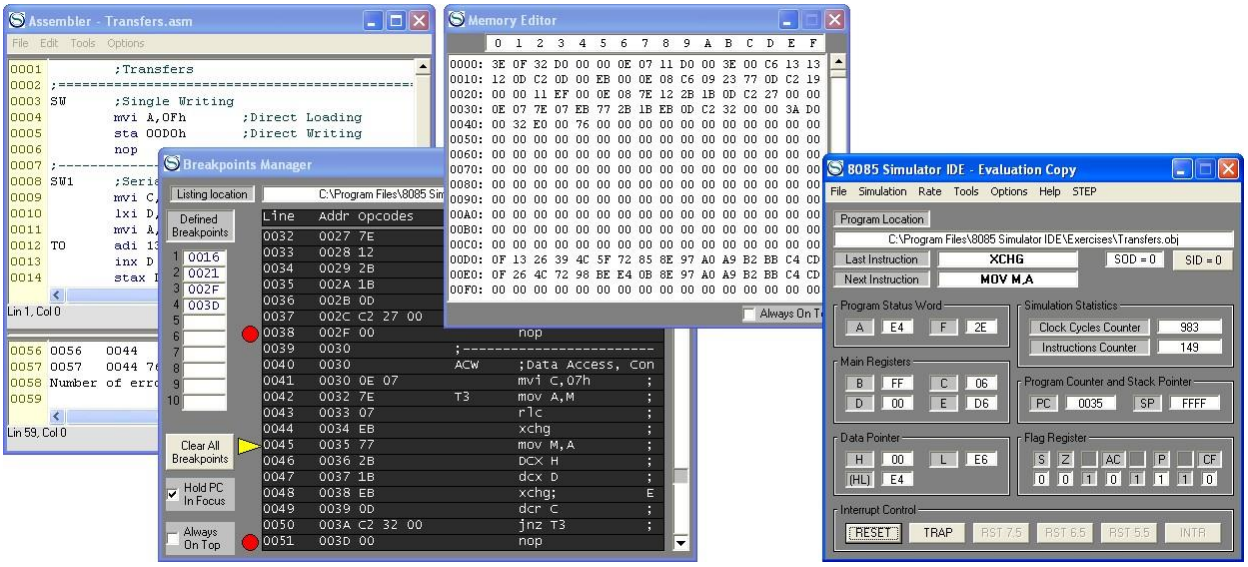
Бұл тарауда ұсынылған бағдарлама әртүрлі (мөлшері мен сілтеу тәсіліне қарай) дерек тарату командаларының орындалуы кезіндегі іс-әрекеттермен таныстыруға арналған.

Бірінші блокта (Single Writing) жадыға жеке дерек жіберілімі тура сілтемелі команда арқылы жүзеге асырылады. Екінші блокта (Serial Writing-1) деректер массивінің (арифметикалық прогрессия мүшелері түріндегі) жадыға жанама сілтеу командасымен жіберілуі жүзеге асырылады. Деректер массивін жіберудің одан ұтымдырақ тәсілі келесі блокта (Single Writing-2) келтірілген.

Келесі үш блокта (Data Access, Data Access and Conversion, Single Writing) жадының бір бөлігіндегі деректердің оның екінші бөлігіне жіберілуі алдыңғы блокта пайдаланылған командалар сияқты командалармен (кері тәртіппен) жүзеге асырылады.

Симулятормен жұмыс істеу келесі тәртіппен жүзеге асырылады:

– симулятор бағдарламасын әлпеттемесімен ашыңыз да, ашылған симулятор терезесінде (5.1 сурет) Tools/Assembler таңдап, кезекті Assembler терезесінде File/New таңдаңыз;



5.1 сурет

– бағыттаманы Tab түймесімен ығыстырып, төмендегі бағдарламаның бірінші командасын енгізіңіз де, Enter түймесін басыңыз. Осы сияқты бағдарламаның келесі командаларын да енгізіңіз. Белгілер жол басында жазылады.

– енгізілген бағдарламаны Data Transfers (Дерек жіберілімдері) аталымымен сақтаңыз (File/Save As), нәтижесінде asm кеңейтілімді файл құрылады. Сосын, Assembler терезесінде Tools/Assemble&Load таңдаңыз, нәтижесінде hex, lst, obj кеңейтілімді файлдар құрылады;

– симулятор терезесінде Tools/Peripheral Devices терезесін таңдаңыз. Ашылған Peripheral Devices терезесінде Device 1 түймесін басып, бұл порттың адресін 01 ретінде анықтап, сосын осы портты IN жағдайына қойыңыз. Осы сияқты келесі порттардың адресін 02, 03 және 04 ретінде анықтап, оларды OUT жағдайына қойыңыз;

– симулятордың негізгі терезесінде Tools/Memory Editor арқылы жады терезесін ашыңыз;

– симулятордың негізгі терезесінде қадамдап моделдеу режимін (Rate/Step By Step) таңдап, симуляторды іске қосыңыз (Simulation/Start);

– Step командасын беріп, микропроцессор регистрлері мен сәйкесті жады ұяшықтарындағы өзгерістерді бақылау арқылы бағдарламаның орындалу нәтижелерін талдаңыз;

– бағдарлама орындалып болғаннан кейін, File/Load Program командасымен бағдарламаны қайта енгізіңіз, нәтижесінде жады терезесі жаңартылады, яғни онда команда кодтары ғана қалады;

– симулятордың негізгі терезесінде Tools/Breakpoints Manager таңдап, әрбір блоктан кейін, яғни келесі блоктың бірінші командаларына тексерім нүктелерін қойыңыз;

– симулятордың негізгі терезесінде модельдеу жылдамдығын (мысалы, Rate/Normal) таңдап, симуляторды іске қосыңыз (Simulation/Start);

– бағдарламаның әрбір тексерім нүктесіндегі тоқталымында орындалған бағдарлама блогындағы командалардың іс-әрекетін және алынған нәтижелерді (микропроцессор регистрлері мен жадының сәйкесті ұяшықтарындағы) талдаңыз.

Бағдарлама мәтіні төменде келтірілген:

```
;*****************************************************
;                       Data Transfers
;*****************************************************
;                       Writing to Memory
;=====================================================
;                       Single Data
;-----------------------------------------------------------------------
                MVI     A,0Fh
                OUT     02h
                STA     00D0h
;=====================================================
;                       Data Array-1
;-----------------------------------------------------------------------
                MVI     C,07h
                LXI     D,00D0h
                IN      01h             ;Initial Data
DT_1:           ADI     13h             ;Step
                OUT     02h
                INX     D
                STAX    D
                DCR     C
                JNZ     DT_1
;=====================================================
;                       Data Array-2
;-----------------------------------------------------------------------
                XCHG
                MVI     C,08h
```

```
DT_2:   ADI     09h     ;Step
        OUT     02h
        INX     H
        MOV     M,A
        DCR     C
        JNZ     DT_2
;*************************************************
;                       Data Access
;=================================================
;                       Conversion-1
;-------------------------------------------------------------------------
        LXI     D,00EFh
        MVI     C,08h
CNV_1:  MOV     A,M
        OUT     03h
        RRC
        OUT     04h
        STAX    D
        DCX     H
        DCX     D
        DCR     C
        JNZ     CNV_1
;=================================================
;                       Conversion-2
;-------------------------------------------------------------------------
        MVI     C,07h
CNV_2:  MOV     A,M
        OUT     03h
        RAR
        OUT     04h
        XCHG
        MOV     M,A
        DCX     H
        DCX     D
        XCHG
        DCR     C
        JNZ     CNV_2
;=================================================
;                       Single Access
;-------------------------------------------------------------------------
        LDA     00D0h
        OUT     03h
        CMA
        OUT     04h
```

```
            STA     00E0h
;=============================================
            HLT
;---------------------------------------------------------------------
```

5.2 Бірбайтты арифметика

Келесі бағдарлама (Single-Byte Arithmetic) бірбайтты деректермен жұмыс атқарушы, әртүрлі (мөлшері мен сілтеу тәсілі бойынша) арифметикалық командалардың орындайтын іс-әрекеттерін түсіндіруге арналған.

Entry Data блогында бастапқы параметрлер (n – мүшелер саны, d – прогрессияның өсім қадамы, a0 – бастапқы мәні) енгізіледі де, олардың негізінде Data Array блогында (арифметикалық прогрессия мүшелерінің мәндері түрінде) деректер қалыптастырылып, олар жадыға жазылады.

Cumulative Summation блогында арифметикалық прогрессия мүшелерінің алдыңғы блокта жадыға жазылған мәндерінің қосындысы есептеледі.

Direct Summation блогында тіке сілтемді қосу командасы орындалады.

Алынған қосу нәтижесін жадыға сақтау Result of Summation блогында жүзеге асырылады.

Series Subtraction және Direct Subtraction блоктарында қосуға пайдаланылған деректердің алдыңғы алынған қосу нәтижесінен тізбелеп алыну процесі жүзеге асырылады.

Бағдарлама мәтіні төменде келтірілген:

```
;*****************************************************
;                   Single-Byte Arithmetic
;*****************************************************
;                   Entry Data (for Linear Series)
;---------------------------------------------------------------------
            LXI     H,00D0h
            IN      01h         ;Data Quantity
            MOV     C,A
            MOV     M,A
            IN      01h         ;Step
            MOV     D,A
            IN      01h         ; Initial Data
;=============================================
;                   Data Array (Linear Series)
;---------------------------------------------------------------------
DT:         ADD     D
            OUT     02h
            INX     H
            MOV     M,A
```

```
            DCR     C
            JNZ     DT
;=============================================
;                   Cumulative Summation
;---------------------------------------------------------------------
            LXI     H,00D0h
            MOV     C,M
            MVI     A,00h
            MOV     B,A
SUM:        INX     H
            ADD     M
            JNC     NC_M
            INR     B
;.....................................................................
NC_M:       OUT     03h
            MOV     E,A
            MOV     A,B         ;Temporary
            OUT     04h
            MOV     A,E         ;Recovery
;.....................................................................
            DCR     C
            JNZ     SUM
;=============================================
;                   Direct Addition
;---------------------------------------------------------------------
            ADI     0E5h
            JNC     NC_D
            INR     B
;.....................................................................
NC_D:       OUT     03h
            MOV     E,A
            MOV     A,B
            OUT     04h
            MOV     A,E
;=============================================
;                   Result of Summation
;---------------------------------------------------------------------
            LXI     H,00F0h
            MOV     M,A
            INX     H
            MOV     M,B
;*****************************************************
;                   Series Subtraction
;---------------------------------------------------------------------
```

```
            LXI     H,00F0h
            MOV     A,M
            INX     H
            MOV     B,M
;……...
            LXI     H,00D0h
            MOV     C,M
SUB:        INX     H
            SUB     M
            JNC     NB_M
            DCR     B
;……...
NB_M:       OUT     03h
            MOV     E,A
            MOV     A,B
            OUT     04h
            MOV     A,E
;……...
            DCR     C
            JNZ     SUB
;=============================================
;                   Direct Subtraction
;……...
            SUI     0E5h
            JNC     NB_D
            DCR     B
;……...
NB_D:       OUT     03h
            MOV     E,A
            MOV     A,B
            OUT     04h
            MOV     A,E
;=============================================
            HLT
;-------------------------------------------------------------------------
```

Бұл бағдарламаны да алдында суреттелген тәртіппен, яғни алдымен қадамдау режимінде (Rate/Step By Step), сосын жылдам режимде (Rate/Normal) зерттеуге болады.

5.3 Екібайтты арифметика

Бұл тарауда келтірілген бағдарлама бірбайтты деректермен жұмыс атқарушы, әртүрлі (мөлшері мен сілтеу тәсілі бойынша) арифметикалық командалар (қосу және алу) арқылы екібайтты деректермен сәйкесті

операциялардың орындалуын жүзеге асыруға пайдаланылу кезіндегі іс-әрекеттерін түсіндіруге арналған.

Entry Data блогында бастапқы параметрлер (n – мүшелер саны, d – прогрессияның өсім қадамы, a0 – бастапқы мәні) енгізіледі де, олардың негізінде Data Array блогында (арифметикалық прогрессия мүшелерінің екібайтты мәндері түрінде) деректер қалыптастырылып, олар жадыға жазылады.

Cumulative Summation блогында арифметикалық прогрессия мүшелерінің алдыңғы блокта жадыға жазылған мәндерінің қосындысы есептеледі.

Direct Summation блогында тіке сілтемді командалар арқылы екібайтты деректер қосу операциясы орындалады.

Алынған қосу нәтижесін жадыға сақтау Result of Summation блогында жүзеге асырылады.

Series Subtraction және Direct Subtraction блоктарында қосуға пайдаланылған екібайтты деректердің алдыңғы алынған қосу нәтижесінен тізбелеп алыну процесі жүзеге асырылады.

Бағдарлама мәтіні төменде келтірілген:

```
        ;*****************************************************
                        Data Transfers
;*****************************************************
;                       Writing to Memory
;=============================================
;                       Single Data
;---------------------------------------------------------------------
            MVI     A,0Fh
            OUT     02h
            STA     00D0h
;=============================================
;                       Data Array-1
;---------------------------------------------------------------------
            MVI     C,07h
            LXI     D,00D0h
            IN      01h         ;Initial Data
DT_1:       ADI     13h         ;Step
            OUT     02h
            INX     D
            STAX    D
            DCR     C
            JNZ     DT_1
;=============================================
;                       Data Array-2
;---------------------------------------------------------------------
```

```
            XCHG
            MVI     C,08h
DT_2:       ADI     09h         ;Step
            OUT     02h
            INX     H
            MOV     M,A
            DCR     C
            JNZ     DT_2
;*************************************************
;                   Data Access
;=================================================
;                   Conversion-1
;-------------------------------------------------------------------------
            LXI     D,00EFh
            MVI     C,08h
CNV_1:      MOV     A,M
            OUT     03h
            RRC
            OUT     04h
            STAX    D
            DCX     H
            DCX     D
            DCR     C
            JNZ     CNV_1
;=================================================
;                   Conversion-2
;-------------------------------------------------------------------------
            MVI     C,07h
CNV_2:      MOV     A,M
            OUT     03h
            RAR
            OUT     04h
            XCHG
            MOV     M,A
            DCX     H
            DCX     D
            XCHG
            DCR     C
            JNZ     CNV_2
;=================================================
;                   Single Access
;-------------------------------------------------------------------------
            LDA     00D0h
            OUT     03h
```

```
            CMA
            OUT         04h
            STA         00E0h
;=========================================
            HLT
;-------------------------------------------------------------------------
```

5.4 Стекпен жұмыс ұйымдастырылуы

Бұл тарауда келтірілген бағдарлама стек пен микропроцессор регистрлері арасындағы дерек жіберілімдерін ұйымдастыру принциптерін және екібайтты деректерді қосу операцияларын DAD командасы арқылы жүзеге асыру тәртібін түсіндіруге арналған.

Entry Data блогында бастапқы параметрлер (n – мүшелер саны, d – прогрессияның өсім қадамы, a0 – бастапқы мәні) енгізіледі де, олардың негізінде Data Array блогында (арифметикалық прогрессия мүшелерінің екібайтты мәндері түрінде) деректер қалыптастырылып, олар жадыға жазылады.

Cumulative Summation блогында арифметикалық прогрессия мүшелерінің алдыңғы блокта жадыға жазылған мәндерінің қосындысы есептеледі.

Direct Summation блогында тіке сілтемді командалар арқылы екібайтты деректер қосу операциясы орындалады.

Бағдарлама мәтіні төменде келтірілген:

```
;*****************************************************
;                       Stack
;*****************************************************
;                       Entry Data (for Linear Series)
;-------------------------------------------------------------------------
            IN          01h             ;Data Quantity
            MOV         B,A
            IN          01h             ;Step (Low Byte)
            MOV         E,A
            IN          02h             ;Step (High Byte)
            MOV         D,A
            IN          01h             ;Initial Data (Low Byte)
            MOV         L,A
            IN          02h             ; Initial Data (High Byte)
            MOV         H,A
;=========================================
;                       Data Array (Linear Series)
;-------------------------------------------------------------------------
            LXI         SP,00B0h        ;Stack Pointer
;.........................................................................
```

```
            MOV     C,B
DT:         DAD     D
;.....................................................................................
            MOV     A,L
            OUT     03h
            MOV     A,H
            OUT     04h
;.....................................................................................
            PUSH    H
            DCR     C
            JNZ     DT
;=======================================
;                   Cumulative Summation
;-------------------------------------------------------------------
            MOV     C,B
            LXI     H,0000h     ;Doublet-Byte Accumulator
SUM:        POP     D
            DAD     D
;.....................................................................................
            MOV     A,L
            OUT     03h
            MOV     A,H
            OUT     04h
;.....................................................................................
            DCR     C
            JNZ     SUM
;=======================================
;                   Direct Addition
;-------------------------------------------------------------------
            LXI     D,2C1Dh
            DAD     D
;.....................................................................................
            MOV     A,L
            OUT     03h
            MOV     A,H
            OUT     04h
;=======================================
            HLT
;-------------------------------------------------------------------
```

5.5 Көбейту операциясының жүзеге асырылуы

Келтірілген бірбайтты екілік сандарды көбейту бағдарламасы матрицалы көбейту алгоритмі бойынша құрылған:

– 8-разрядты сандарды көбейту нәтижесі 16-разрядты сан болатындықтан және жеке қосылымдардың да 16-разрядты сандарға жүргізілетіндігіне байланысты, 16-разрядты аккумулятор ретінде H регистрлік жұбы алынып, бағдарламаның жұмыс басталымында ол тазартылады;

– D регистрлік жұбының E регистріне бірінші сан (көбейтілгіш) енгізіліп, ал D регистрі тазартылады да, нәтижесінде көбейтілгіш 16-разрядты санға айналдырылады;

– ығыстыру мүмкіндігін қамтамасыз ету үшін, екінші сан (көбейткіш) A регистрінде (8-разрядты аккумуляторда) орналастырылады;

– ығыстыру операциясының санын бақылауды қамтамасыз ету үшін, көбейткіштің разряд саны (8) C регистріне енгізіледі;

– әрбір жеке көбейтілім алдында, одан бұрын жиналған нәтиже (H регистрлік жұбында) солға ығыстырылады (еселеу арқылы);

– көбейткіштің кезекті разрядының мәні A регистрінің құрамын (көбейткішті) солға ығыстыру арқылы анықталып, оның нөлге тең жағдайында – жеке көбейтілім осымен-ақ (яғни, H регистрлік жұбының алдыңғы ығыстырылымымен) бітеді де, ал оның бірге тең жағдайында – H регистрлік жұбының құрамына D регистрлік жұбының құрамы қосылады;

– әрбір жеке көбейтілімнен кейін ығыстырылым санауышының құрамы бірге кемітіледі де, оның нөлге теңескенінде – көбейту бағдарламасы өз жұмысын аяқтайды.

Көбейту операциясы жүргізілетін бірбайтты сандардың енгізілуі мен көбейту нәтижесінің көрсетімге шығарылуы деректер енгізу/шығару жұмысын атқаруға арналған шеткі құрылғы арқылы жүзеге асырылады.

Бағдарлама мәтіні төменде келтірілген:

```
;*****************************************************
;                    Multiplication
;*****************************************************
;                    Preparation
;-----------------------------------------------------------------
            IN      01h         ;Multiplicand
            MOV     E,A
            MVI     D,00h
            IN      02h         ;Multiplier
            LXI     H,0000h     ;Result
            MVI     C,08h       ;Digit Counter
;=================================================
;                    Implementation
;-----------------------------------------------------------------
SH:         DAD     H           ;Double-Byte Shift
            RLC
            JNC     NC
            DAD     D           ;Double-Byte Addition
```

```
NC:     DCR     C
        JNZ     SH
;……..
        MOV     A,L
        OUT     03h
        MOV     A,H
        OUT     04h
;=============================================
        HLT
;-------------------------------------------------------------------------
```

5.6 Қосалқы бағдарламалармен жұмыс ұйымдастырылуы

Бұл тарауда келтірілген қосалқы бағдарламалармен жұмыс ұйымдастырылуы бағдарламасы келесі тәртіппен құрылған. Негізгі бағдарламаның басында, стек көрсеткішіне (SP) нақтылы сан (адрес) енгізу арқылы, жадының бір бөлігі стекке бөлінеді.

Mul1 және Mul2 блоктарының әрбіреуінде көбейтуге арналған деректер енгізіледі де, көбейту операциясын орындауға арналған қосалқы бағдарлама (MUL), сосын нәтиже көрсетуші қосалқы бағдарлама (PER) шақырылады.

Sum0 блогында қосу операцияларын орындауға деректер (арифметикалық прогрессияның параметрлері) енгізіледі де, қосу операцияларын орындауға арналған қосалқы бағдарлама (SUM), сосын нәтиже көрсетуші қосалқы бағдарлама (PER) шақырылады.

Негізгі бағдарламаның Sum=... блогында ақырғы нәтиже анықталады, яғни алдыңғы кезеңдерде алынған және жадының стек бөлігінде сақталған, жеке нәтижелердің қосындысы есептеледі.

Бағдарламада үш қосалқы бағдарлама пайдаланылады:

1) Multiplication – қосалқы бағдарламаның шақырулар алдында енгізілген екі санды көбейтеді.

2) Cumulative Summation – қосалқы бағдарламаның шақырылар алдында параметрлері берілген арифметикалық прогрессияның мүшелерінің мәндерінің қосындысын анықтайды.

3) Displaying – қосалқы бағдарламаларда алынған нәтижелерді шеткі құрылғыда көрсетіп, оларды стекте сақтайды.

Бағдарлама мәтіні төменде келтірілген:

```
;*****************************************************
;                       Composite Program
;*****************************************************
;                        SUM=a1*a2+a3*a4+sum(ai,n,d)
;-------------------------------------------------------------------
        LXI     SP,00FFh
;=============================================
;                        Mul1=a1*a2
```

```
;-------------------------------------------------------------------
            IN          01h             ;Multiplicand
            MOV         E,A
            IN          02h             ;Multiplier
            CALL        MUL
            PUSH        H
            CALL        PER
;===================================================
;                       Mul2=a3*a4
;-------------------------------------------------------------------
            IN          01h             ;Multiplicand
            MOV         E,A
            IN          02h             ;Multiplier
            CALL        MUL
            PUSH        H
            CALL        PER
;===================================================
;                       Sum0(ai,n,d)
;-------------------------------------------------------------------
            IN          01h             ;Data Quantity - n
            MOV         C,A
            IN          02h             ;Step - d
            MOV         D,A
            IN          01h             ; Initial Operand – a0
            MOV         E,A
            CALL        SUM
            PUSH        H
            CALL        PER
;===================================================
;                       Sum=Mul1+Mul2+Sum0
;-------------------------------------------------------------------
            POP         H
            POP         D
            DAD         D
            POP         D
            DAD         D
            CALL        PER
;===================================================
            HLT
;***************************************************
MUL:                    ;Multiplication (subroutine)
;-------------------------------------------------------------------
            MVI         D,00h
            LXI         H,0000h
```

```
        MVI     C,08h
ML:     DAD     H
        RLC
        JNC     NC
        DAD     D
NC:     DCR     C
        JNZ     ML
        RET
;=============================================
SUM:            ; Cumulative Summation (subroutine)
;-----------------------------------------------------------------------
        LXI     H,0000h
SM:     MOV     A,E
        ADD     D
        MOV     E,A
        MOV     A,L
        ADD     E
        JNC     NCS
        INR     H
NCS:    MOV     L,A
        DCR     C
        JNZ     SM
        RET
;=============================================
PER:            ; Displaying
;-----------------------------------------------------------------------
        MOV     A,L
        OUT     03H
        MOV     A,H
        OUT     04H
        RET
;-----------------------------------------------------------------------
```

5.7 Екілік-ондық түрлендірілімдер

Ұсынылған бағдарламада бұрын қарастырылған көбейту бағдарламасына екілік-ондық түрлендірілімін жүзеге асыру бағдарламасы қосылып, ол арқылы енгізілген бірбайтты көбейткіштерге де, екібайтты нәтижеге де сәйкесті түрлендірілім жүргізіледі.

Ондық көрсетілімдегі сандардың сәйкесті разрядтарының мәні түрлендірілетін санды ондық санау жүйесінің сәйкесті разрядының құнына бөлу арқылы анықталады. Intel 8085 микропроцессорының командалар жүйесінде бөлу командасы болмағандықтан, бөлу операциясы алу командалары арқылы жүзеге асырылады.

Түрлендірілетін сандардың оналтылық мәндері мен сәйкесті түрлендірілім нәтижелері жадының бөлінген ұяшықтарына шығарылады. Түрлендірілім нәтижелерін бақылаудың көрнекілігін қамтамасыз ету мақсатында, олар құрамы FFh түріндегі ұяшықтар қоршамында шығарылады.

Бағдарлама мәтіні төменде келтірілген:

```
;*************************************************
;                       BinDec
;*************************************************
;                       Setup Cleaning
;-------------------------------------------------------------------------
            LXI     H,00D0h
            MVI     C,2Fh
            MVI     A,00h
CLR         MOV     M,A
            INX     H
            DCR     C
            JNZ     CLR
;=========================================
;                       Multiplication
;-------------------------------------------------------------------------
            IN      01h         ;Multiplicand
;.........................................................................................
            LXI     H,00D0h
            MOV     M,A
            INX     H
            MVI     M,00h
;.........................................................................................
            MOV     E,A
            MVI     D,00h
            IN      02h         ;Multiplier
;.........................................................................................
            LXI     H,00E0h
            MOV     M,A
            INX     H
            MVI     M,00h
;.........................................................................................
            LXI     H,0000h
            MVI     C,08h
;=============================================
SH:         DAD     H
            RLC
            JNC     NC
            DAD     D
```

```
NC:     DCR     C
        JNZ     SH
;-------------------------------------------------------------------------
        MOV     A,L
        OUT     03h
        MOV     A,H
        OUT     04h
;.........................................................................
        XCHG
        LXI     H,00F0h
        MOV     M,E
        INX     H
        MOV     M,D
;=========================================
;               Preparation to BinDec
;=========================================
        LXI     H,00D0h
        MOV     E,M
        INX     H
        MOV     D,M
        LXI     H,00D8h
        CALL    BD
;-------------------------------------------------------------------------
        LXI     H,00E0h
        MOV     E,M
        INX     H
        MOV     D,M
        LXI     H,00E8h
        CALL    BD
;-------------------------------------------------------------------------
        LXI     H,00F0h
        MOV     E,M
        INX     H
        MOV     D,M
        LXI     H,00F8h
        CALL    BD
;-------------------------------------------------------------------------
        HLT
;*****************************************************
BD:             ;BinDec
;-------------------------------------------------------------------------
        MVI     M,0FFh
;-----------------------------------------
Dig4:   DCX     H
```

```
D4:     MOV     A,E
        SUI     10h         ;Dec(10000)=Hex(2710)
        MOV     C,A
        MOV     A,D
        SBI     27h
        MOV     B,A
        JC      Dig3
;-------------------------------------------------------------------------------
        INR     M
        MOV     E,C
        MOV     D,B
        JMP     D4
;-------------------------------------------------------------------------------
Dig3:   DCX     H
D3:     MOV     A,E
        SUI     0E8h        ;Dec(1000)=Hex(03E8)
        MOV     C,A
        MOV     A,D
        SBI     03h
        MOV     B,A
        JC      Dig2
;-------------------------------------------------------------------------------
        INR     M
        MOV     E,C
        MOV     D,B
        JMP     D3
;-------------------------------------------------------------------------------
Dig2:   DCX     H
D2:     MOV     A,E
        SUI     64h         ;Dec(100)=Hex(0064)
        MOV     C,A
        MOV     A,D
        SBI     00h
        MOV     B,A
        JC      Dig1
;-------------------------------------------------------------------------------
        INR     M
        MOV     E,C
        MOV     D,B
        JMP     D2
;-------------------------------------------------------------------------------
Dig1 :  DCX     H
D1:     MOV     A,E
        SUI     0Ah         ;Dec(10)=Hex(0A)
```

```
                MOV     C,A
                JC      D0
;-------------------------------------------------------------------------
                INR     M
                MOV     E,C
                JMP     D1
;-------------------------------------------------------------------------
D0              DCX     H
                MOV     M,E
;-------------------------------------------------------------------------
                DCX     H
                MVI     M,0FFh
;=========================================================
                RET
;=========================================================
```

5.8 Үзілістермен жұмыс ұйымдастырылуы

Бұл тараудағы бағдарламаның құрылымын былай суреттеуге болады. Жадының 0050h ұяшығынан бастап орналасқан негізгі бағдарламада екібайтты сандардың тізбелі қалыптастырылуы және олардың шеткі құрылғының 1 және 2 порттарында шығарылуы жүзеге асырылады.

Үзіліс бағдарламаларында логикалық командалар арқылы, шеткі құрылғының 3 және 4 порттарында шығарылатын, динамикалы көріністер тудырылады.

Бағдарлама мәтіні төменде келтірілген:

```
;*********************************************************
;                       Interrupts
;*********************************************************
                JMP     0050h       ;Jump to Main Routine
;=========================================================
                .ORG    002Ch       ;Interrupt Routine 5.5
;-------------------------------------------------------------------
                JMP     Rst55
;=========================================================
                .ORG    003Ch       ;Interrupt Routine 7.5
;-------------------------------------------------------------------
                JMP     Rst75
;*********************************************************
                .ORG    0050H       ;Main Rountine
;-------------------------------------------------------------------
                LXI     SP,00FFh
                MVI     A,00h
                MOV     B,A
```

```
                OUT     01h
                OUT     02h
;-------------------------------------------------------------------------
MR0:    MVI     A,00h
MR:     INR     A
                OUT     01h
                JNZ     MR
                INR     B
                MOV     A,B
                OUT     02h
;.........................................................................
                RIM                     ;Read Interrupt Mask
;.........................................................................
                MVI     A,0BH           ;Interrupt Mask
                SIM                     ;Set Interrupt Mask
                EI                      ;Enable Interrupts
;.........................................................................
                RIM
;.........................................................................
                JNZ     MR0
                DI                      ;Disable Interrupts
;.........................................................................
                RIM
;.........................................................................
                JMP     MR0
                HLT
;*********************************************************
;=========================================
Rst55:                  ;Interrupt Routine 5.5
;-------------------------------------------------------------------------
                PUSH    PSW
                PUSH    B
;-------------------------------------------------------------------------
                MVI     E,00h
                MVI     B,08h
                MVI     A,00h
                OUT     04h
Int5_0:         MOV     C,B
                MVI     A,80h
Con5_0:         RLC
                MOV     H,A
                ORA     E
                MOV     L,A
                OUT     04h
```

```
                MOV     A,H
                DCR     C
                JNZ     Con5_0
                DCR     B
                MOV     E,L
                JNZ     Int5_0
;-------------------------------------------------------------------------
                MVI     B,08h
Int5_1:         MOV     C,B
                MVI     A,07Fh
Con5_1:         RLC
                MOV     H,A
                ANA     E
                MOV     L,A
                OUT     04h
                MOV     A,H
                DCR     C
                JNZ     Con5_1
                DCR     B
                MOV     E,L
                JNZ     Int5_1
;……...
                POP     B
                POP     PSW
;……...
                RET                     ;Return from Interrupt
;=========================================
Rst75:          ;Interrupt Routine 7.5
;-------------------------------------------------------------------------
                PUSH    PSW
                PUSH    B
;……...
                MVI     E,00h
                MVI     B,08h
                MVI     A,00h
                OUT     03h
Int7_0:         MOV     C,B
                MVI     A,01h
Con7_0:         RRC
                MOV     H,A
                ORA     E
                MOV     L,A
                OUT     03h
                MOV     A,H
```

```
                DCR     C
                JNZ     Con7_0
                DCR     B
                MOV     E,L
                JNZ     Int7_0
;-------------------------------------------------------------------------
                MVI     B,08h
Int7_1:         MOV     C,B
                MVI     A,0FEh
Con7_1:         RRC
                MOV     H,A
                ANA     E
                MOV     L,A
                OUT     03h
                MOV     A,H
                DCR     C
                JNZ     Con7_1
                DCR     B
                MOV     E,L
                JNZ     Int7_1
;……...
                POP     B
                POP     PSW
;……...
                RET                     ;Return from Interrupt
;-------------------------------------------------------------------------
```

6 Микропроцессорлық жүйенің моделі

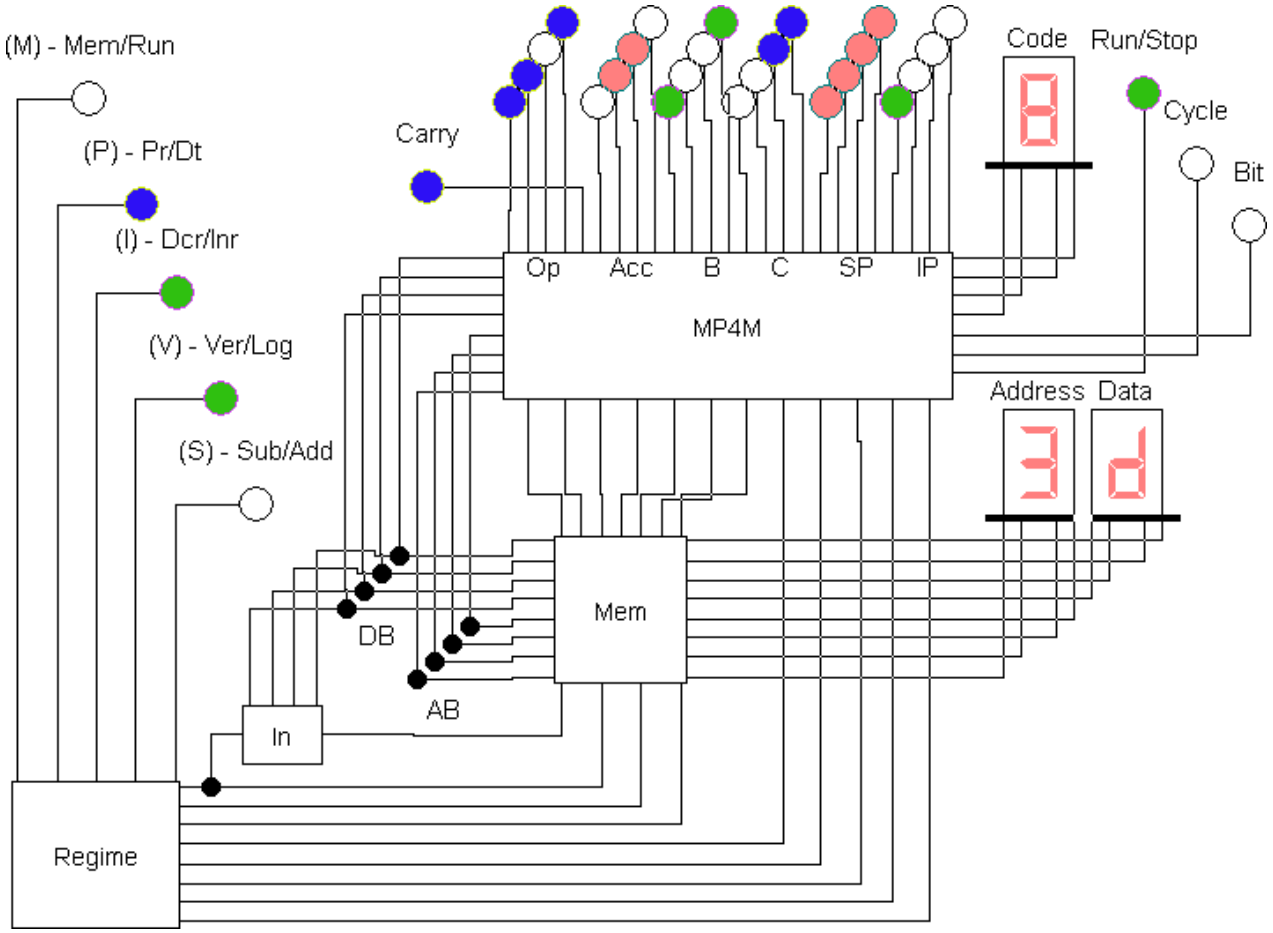
Осы тараудағы материал микропроцессорлық жүйенің жалпылама құрылым принциптерін жүзеге асыру мүмкіндіктерін, ерекше командалар жүйесімен қамтылған, осындай жүйенің моделін Electronics Workbench модельдеу жүйесінде [2] құру арқылы қарастыруға арналған.

Қарастырылған модель оқырмандарға микропроцессорлық жүйенің құрылым принциптері мен олардың командалар жүйесі туралы толық білім алуға және бұндай жүйелерді құру жолдарын үйренуге мүмкіндік береді.

Модельдің қарапайымдылығы мен оның жұмысының көрнекілігін қамтамасыз ету мақсатында, ол 4-разрядты деректермен (тетрадалармен) жұмыс істеугеесептеліп, микропроцецессорлық жалпы құрылым принциптері бойынша құрылды.

6.1 MP4M моделінің құрылымы

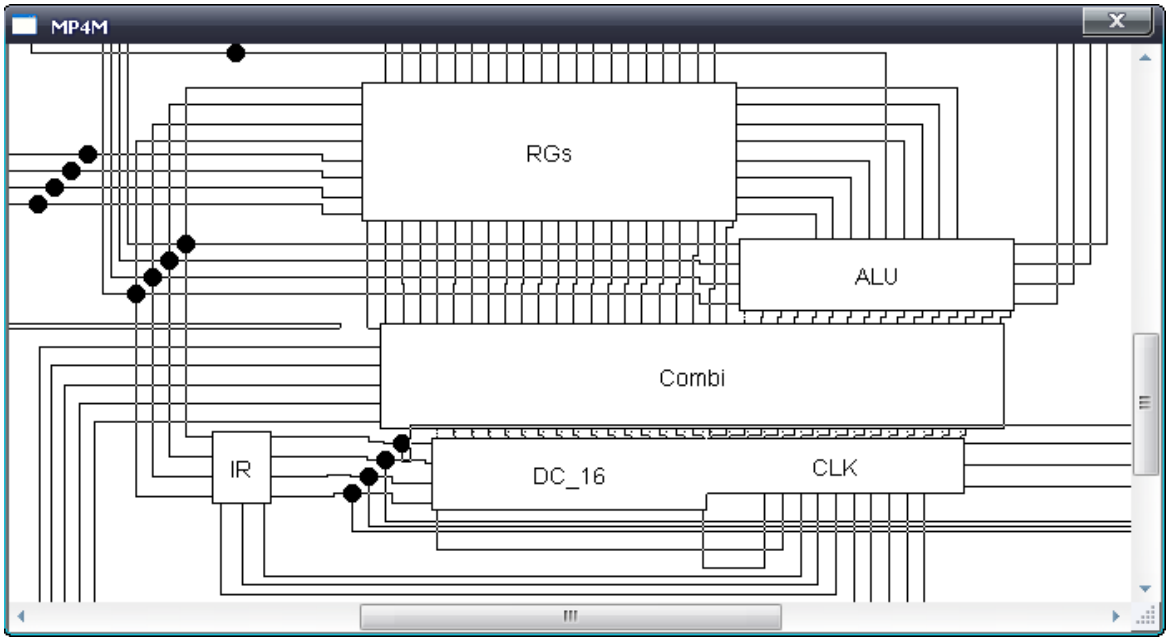
Құрамына микропроцессор (MP4M), дерек енгізу құрылғысы (In), жады жүйесі (Mem) және жүйенің жұмыс режимін таңдау блогы (Regime) енгізілген, 4-разрядты микропроцессорлық жүйенің моделі 6.1 суретте келтірілген.



6.1 сурет

Микропроцессор құрамына (6.2 сурет) келесі блоктар кіреді:

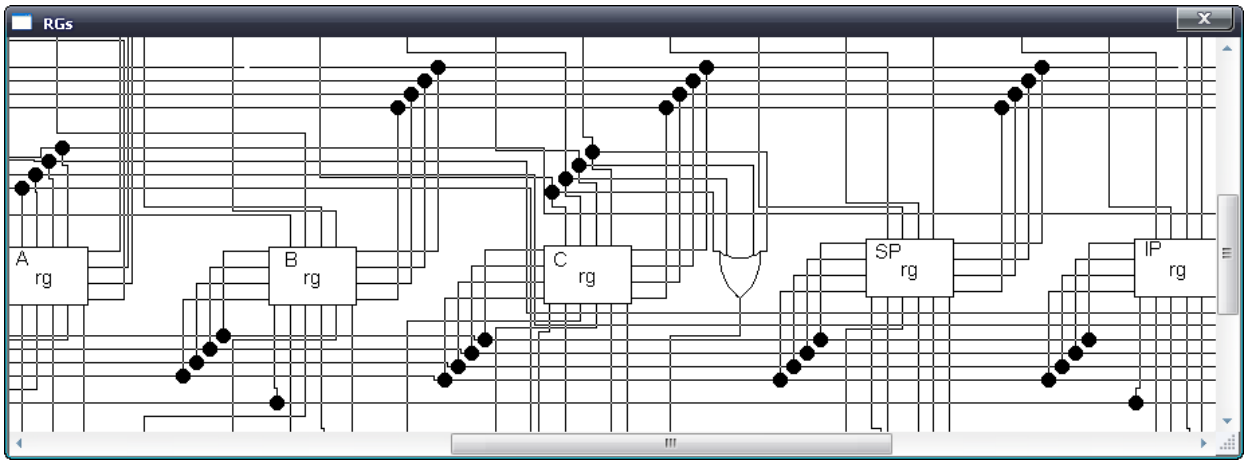
- арнайы қызметтік регистрлер жинамы (RGs, Registers);
- арифметикалық-логикалық құрылғы (ALU, Arithmetic-Logic Unit);
- команда регистрі (BUF, Buffer Register);
- команда дешифраторы (DC_16, Decoder 16-bit);
- таймер (CLK, Clock);
- микробағдарламалы құрылғы (Combi, Combinational Circuit.



6.2 сурет

6.2 MP4M моделінің қызмет буындары

Регистрлер жинамының (RGs) құрамына әрбіреуі нақтылы қызмет атқаруға арналған келесі регистрлер кіреді:



6.3 сурет

– A регистрі аккумулятор (Accumulator) қызметін атқарады, онда арифметикалық және логикалық операцияларға қатысты операндтардың біреуі немесе операциялардың орындалу нәтижесі сақталады. Аккумуляторға дерек жадыдан (LD A командасы) немесе ALU-дан (кезекті арифметикалық немесе логикалық операция орындалғаннан кейін) енгізіледі;

– A1 регистрі айналымды қосылымдардың ұйымдастырылуы кезінде туатын тасымалдарды жинауға арналған. Ол бағдарламаның іске қосылуы кезінде және аккумуляторға (A регистріне) LD A командасымен дерек

енгізілуі кезінде тазартылады. Ондағы дерек Space түймесін қысқа уақытқа басу арқылы аккумулятор көрсеткіштеріне шығарылады;

– B регистрі микропроцессордың ішкі регистрлері мен жады арасындағы дерек жіберілімдерінің адрес регистрінің қызметін атқарады (LD C, LD IP, LD SP, LD A командалары). LD C, LD SP, LD A командалары орындалғаннан кейін B регистріндегі сан ұлғайтылады (яғни, оған бір қосылады). Ол бағдарламаның іске қосылуы кезінде тазартылады Бұл регистрге дерек LD B командасымен енгізіледі, оған адрес регистрі ретінде C регистрі пайдаланылады;

– C регистрі арифметикалық және логикалық операцияларға қатысты екінші операндтың адрес регистрінің қызметін атқарады. Сәйкесті командалар орындалғаннан кейін C регистріндегі сан ұлғайтылады, бірақ оның бағытын (B регистрінен ерекше) өзгертуге болады, ол I түймесі арқылы қойылады және оның жағдайы (I) - Dcr/Inr көрсеткішінде көрсетіледі: I=0 кезінде – тура бағытта (Increment), ал I=1 кезінде – теріс бағытта (Decrement). Ондағы санды өзгермеген түрінде де қалдыруға болады, бұндай жағдай туралы жүйенің жұмыс режимдерін суреттеу кезінде айтылады. Бұл регистрге дерек LD C командасымен жүзеге асырылады. Бағдарламаның іске қосылуы кезінде бұл регистр 1111 жағдайына қойылады (XOR элементтері арқылы). Кез келген бағдарламаның құрамында дерек тарату командаларының болатындығынан, бағдарламаның бірінші командасы B регистріне дерек енгізу командасы (LD B) болу керек. C регистрінің сұлбасындағы төрткірісті OR элементі регистрдің нөлдік жағдайын анықтауға арналған, ол арқылы айналымды операциялардың ұйымдастырылуы кезінде шартты ауысу жүзеге асырылады;

– SP регистрі стек көрсеткішінің (Stack Pointer) қызметін атқарады, яғни PUSH және POP командалуы кезінде адрес регистрінің рөлінде болады: ондағы адрес PUSH командасының орындалар алдында кемітіледі (Predecrement) және POP командасы орындалғаннан кейін ұлғайтылады (Postincrement). Бұл регистрге дерек енгізу LD SP командасымен жүзеге асырылады. Бағдарламаның іске қосылуы кезінде бұл регистр 1111 жағдайына қойылады;

– IP регистрі команда көрсеткішінің (Instuction Pointer) қызметін атқарады. Кезекті команданың коды жадыдан шығарылып, команда регистріне сақталғаннан кейін ондағы адрес ұлғайтылады. Бұл регистрге дерек LD IP командасымен жүзеге асырылады. Ондағы адрес, жүйенің кейбір альтернативті жұмыс режимдерінде анықталатын, шартты ауысу кезінде де өзгертіледі. Ол бағдарламаның іске қосылуы кезінде тазартылады.

Барлық регистрдің кірістері – ішкі дерек желісіне, ал шығыстары (үш жағдайлы буферлік элементтер арқылы) атқаратын қызметіне сәйкесті дерек желісіне немесе адрес желісіне қосылған.

Арифметикалық-логикалық құрылғы (ALU, Arithmetic-Logic Unit) арифметикалық және логикалық операциялардың орындалуын жүзеге асырады. Оның құрамында сәйкесті операциялардың орындалуын жүзеге асырушы бөліктемелер (sum, not, and, or, xor) және операндтар мен

орындалған операцияның нәтижесін уақытша сақтауға арналған буферлік регистрлер болады.

Команда дешифраторы (DC_16) кезекті команданың кодына сәйкесті орындаушы микробағдарламасын іске қосады, яғни осы команданың орындалуына қатысты құрылғыларға жіберілетін басқару сигналдарының қалыптастырылу жолын анықтайды.

Таймер (CLK) командалардың орындалуына қатысты құрылғыларға жіберілетін басқару сигналдарының уақыттық таралымын анықтайды.

Қиыстырмалы микробағдарламалық құрылғы (Combi, Combinational Circuit) кезекті команданың орындалуын қамтамасыз етуші кеңістіктік және уақыттық басқару сигналдарын қалыптастырады.

Енгізу құрылғысы кірістеріне сәйкесті түймелер қосылған шифраторлар негізінде құрылған.

Жүйе жадысы 16 ұяшықты екі блок түрінде құрылған. Олардың біреуі– бағдарлама құрамындағы командалардың кодтарын сақтауға, ал екіншісі бағдарламада орындалатын операцияларға қатысты дерек мәндерін және бағдарлама жұмысын басқаруға (мысалы, айналымды бағдарламалы құрылымдар ұйымдастыруға) қажетті ақпарат сақтауға арналған.

Жадыны басқару жүйесі (MCon, Memory Control) жады блоктарының (команда кодтарының немесе деректердің) біреуін және енгізу құрылғысынан немесе микропроцессордан түсетін басқару сигналдарына байланысты олардың жұмыс режимінің (енгізу, жазу, оқу) таңдалуын қамтамасыз етеді.

6.3 MP4M моделінің жұмыс режимдері мен командалар жүйесі

Жүйе бірнеше режимде істей алады, олар режим таңдау блогындағы (Regime) түймелердің жағдайымен анықталады. Олардың жағдайы сәйкесті көрсеткіштерде суреттеледі. Осы режимдерді суреттелік.

Енгізу құрылғысынан жады жүйесіне ақпарат енгізу M=1 кезінде жүргізіледі. Егер бұл кезде P=1 болса, онда ақпарат – бағдарлама жадысына (бағдарлама коды F тоқтату командасымен аяқталу керек), ал P=0 кезінде дерек жадысына енгізіледі. Жады құрамын тексеру Space түймесін қысқа уақытқа басып, жады ұяшығының адресін өзгерту арқылы жүргізіледі. P түймесінің жағдайы өзгертілгенде жады адресі тазартылады.

Бағдарламаны іске қосу M түймесін M=0 жағдайына ауыстыру арқылы жүзеге асырылады. Егер бұл кезде P=1 болса, онда арифметикалық және логикалық командалардың орындалуы кезінде C регистрінің жағдайы өзгертіледі (I мәніне сәйкесті), ал P=0 кезінде оның жағдайы өзгермеген жағдайда қалады.

S=0 кезінде ADS және ACS командалары арқылы – қосу операциялары, ал S=1 – алу операциялары орындалады.

V=0 кезінде AND командасы арқылы екі операнд арасында – қалыпты логикалық көбейту операциясы, ал V=1 кезінде деректің нақтылы разрядындағы бірлік битті табуға арналған биттік операция жүзеге

асырылады. Бұл биттік операцияның орындалуы үшін аккумуляторға қажетті разрядында ғана бір тұрған санауыш енгізілу керек. Суреттелген биттік операция сыртқы құрылғыдан келетін сұраныс сигналын немесе іс жүзіндегі қолданба жүйелердегі құптау сигналын тануды елестетеді.

Жүйенің жұмыс режимдерін ауыстыру бағдарламаны іске қосу алдында жүргізіледі, бірақ оны бағдарламаның жұмыс барысында да жүзеге асыруға болады. Мысалы, жұмыс барысында қосу операцияларын алу операцияларына өзгертуге болады. Сол сияқты конъюнкцияны биттік операцияға ауыстыруға болады.

МР4М микропроцессорының командалары мен олардың сипаттамалары – 6.1 кестеде, ал жүйенің жұмыс режимдерінің қойылу тәртібі 6.2 кестеде келтірілген.

6.1 к е с т е – МР4М микропроцессорының командалары

Мн-ка	Код	Операциялар			
	Hex	V=0	P	S	V=1; I=1
LD B	0	B←[(C)]; C←(C)+1			
LD C	1	C←[(B)]; B←(B)+1			
LD IP	2	IP←[(B)]			
LD SP	3	SP←[(B)]; B←(B)+1			
LD A	4	A←[(B)]; B←(B)+1			
ST	5	[(B)]←(A);B←(B)+1			
INR	6	A←(A)+1			
NOT	7	A← $\overline{(A)}$			
ADS	8	A←(A)±[(C)]; T_C	P=1: C←var(I) P=0: C = invar	S=1: SUB (–) S=0:ADD,ADC (+)	Zc = 0: IP←IP+1
ACS	9	A←(A)±[(C)±T_C; T_C			
AND	A	A←(A)∧[(C)]			Zc∧Za=0: IP←IP+1
OR	B	A←(A)∨[(C)]			
XOR	C	A←(A)$_{\oplus}$[(C)]			
PUSH	D	SP←(SP)–1; (SP)]←(A)			
POP	E	A←(SP)]; SP←(SP)+1			
HLT	F	IP = invar			

6.2 к е с т е – MP4M микропроцессорының жұмыс режимдерінің қойылуы

Кілт, инд-р	M/R = 1		M/R → 0: Running: A,B,IP ← 0; C,SP ← F	
			P=1: C←var(I);	P=0: C = invar
S = 0	P/D = 1: MemPr←IN	P/D = 0: MemD←IN	Addition (+)	A ← (A) + [(C)]
S = 1			Subtraction (–)	A ← (A) – [(C)]
I = 1			PostDecrement C←(C) – 1	C←(C)–1; (C)=0: Zc ← 0
I = 0			PostIncrement C←(C) +1	C ← (C) + 1
V = 0			Logical Operation	A ← (A) {∧ / ∨ / ⊕} [(C)]
V = 1			Verifying of Byte	(A)=0010…b_i = 1: Za ← 0

6.4 MP4M моделінің жұмысын ұйымдастыру мысалдары

Модельдің қызмет мүмкіндіктерін сипаттау мақсатында оның жұмысын ұйымдастырушы екі бағдарламаның суреттемесін келтірелік. Төменде келтірілген бағдарлама қарастырылушы модельдің жады жүйесінде (келтірілген бағдарламада – 1-5 ұяшықтарда) орналастырылған деректердің қосындысын есептеу жұмысын атқарады.

Адрес	Команда	Код	Адрес	Дерек	Адрес	Дерек
0	LD B	0	0	…	…	…
1	LD C	1	1	A	7	5
2	ADS	8	2	B	8	2
3	LD IP	2	3	C	…	…
4	HLT	F	4	D		
			5	E	F	7

Бұл бағдарлама орындалып болғаннан кейін қосу нәтижесінің кіші разрядының мәні A регистрінде (C саны) көрсетіледі, ал оның үлкен разрядының мәні (3 саны) Space түймесін қысқа мерзімге басу арқылы анықталады.

Келесі бағдарлама биттік операцияларды орындау арқылы жадының 1…5 адресті ұяшықтарындағы деректердің ішінен үшінші биті бірлік мәндегі санды табу жұмысын жүзеге асырады.

Адрес	Команда	Код	Адрес	Дерек	Адрес	Дерек
0	LD B	0	0	6	…	…
1	LD C	1	1	8	7	5
2	LD A	4	2	D	8	4
3	AND	A	3	A	9	3
4	LD IP	2	4	9	…	…
5	HLT	F	5	B	F	7

Қажетті сан табылғанда, бағдарлама оны Bit индикаторын жыпылықтату арқылы хабарлайды да, жұмысын бітіреді, ал табылған сан (D саны) A1 регистрінің индикаторларында көрсетіледі.

Қарастырылған модельдің ерекше құрылымы мен оның жұмыс режимдерінің әртүрлілігі мен командалар жүйесінің құбылмалылығы алуан түрлі бағдарламалар құруға мүмкіндік береді және пайдаланушының осы саладағы шығармашылық қабілетінің дамуына әсер етеді.

Әдебиеттер тізімі

1 Угрюмов Е. П. Цифровая схемотехника. – СПб.: БХВ-Петербург, 2010.

2 Кузин А.В., Жаворонков М.А. Микропроцессорная техника. – М.: Академия, 2010.

3 Гусев В.Б. Электроника и микропроцессорная техника. – М.: Высшая школа, 2008.

4 Бойт К. Цифровая электроника. Пер. с нем. – М.: Техносфера, 2007.

5 Опадчий Ю.Ф. Аналоговая и цифровая электроника. – М.: Горячая Линия-Телеком, 2007.

6 Бойко В.И. и др. Схемотехника электронных устройств. Цифровые устройства. – СПб.: БХВ-Петербург, 2004.

7 Бойко В.И. и др. Схемотехника электронных устройств. Микропроцессоры и микроконтроллеры. – СПб.: БХВ-Петербург, 2004.

8 Сугано Т. Дж., Уидмер Н. С. Цифровые системы. Теория и практика: Пер. с англ. – М.: Издательский дом «Вильямс», 2004.

9 Шанаев О.Т. Electronic Workbench моделдеу жүйесі / Қазақ және орыс тілдерінде. – Алматы: 2003.

Мазмұны

Кіріспе....3
1 Цифрлық жүйелердің құрылым негіздері....6
1.1 Логикалық функциялар мен логикалық элементтер....6
1.2 Қиыстырма құрылғылар....10
1.3 Қалыпты қиыстырма құрылғылар....14
1.4 Тізбектеме құрылғылар....21
2 Цифрлық құрылғылардың моделдері....30
2.1 Тізбелі ауыспалы қосу құрылғысы....30
2.2 Алу-қосу құрылғысы....32
2.3 Тізбелі операндтарға арналған қосу құрылғысы....33
2.4 Автоматты режимдегі тізбелі қосу құрылғысы....34
3 Дерек сақтау құрылғылары....35
3.1 RAM түріндегі жады құрылғыларының құрылымдары....37
3.2 ROM түріндегі жады құрылғыларының элементтері....40
4 Микропроцессорлық жүйелер....42
4.1 Микропроцессорлық жүйелердің құрылым принциптері....42
4.2 Intel 8085 микропроцессоры....44
4.3 Intel 8085 микропроцессорының командалар жүйесі....53
4.4 Сілтеу тәсілдері....57
5 Микропроцессорлық жүйелерді бағдарлау....57
5.1 Дерек таратылымдары....58
5.2 Бірбайтты арифметика....62
5.3 Екібайтты арифметика....64
5.4 Стекпен жұмыс ұйымдастырылуы....67
5.5 Көбейту операциясының жүзеге асырылуы....68
5.6 Қосалқы бағдарламалармен жұмыс ұйымдастырылуы....70
5.7 Екілік-ондық түрлендірілімдер....72
5.8 Үзілістермен жұмыс ұйымдастырылуы....76
6 Микропроцессорлық жүйенің моделі....79
6.1 MP4M моделінің құрылымы....80
6.2 MP4M моделінің қызмет буындары....81
6.3 MP4M моделінің жұмыс режимдері мен командалар жүйесі....83
6.4 MP4M моделінің жұмысын ұйымдастыру мысалдары....85
Әдебиеттер тізімі....87