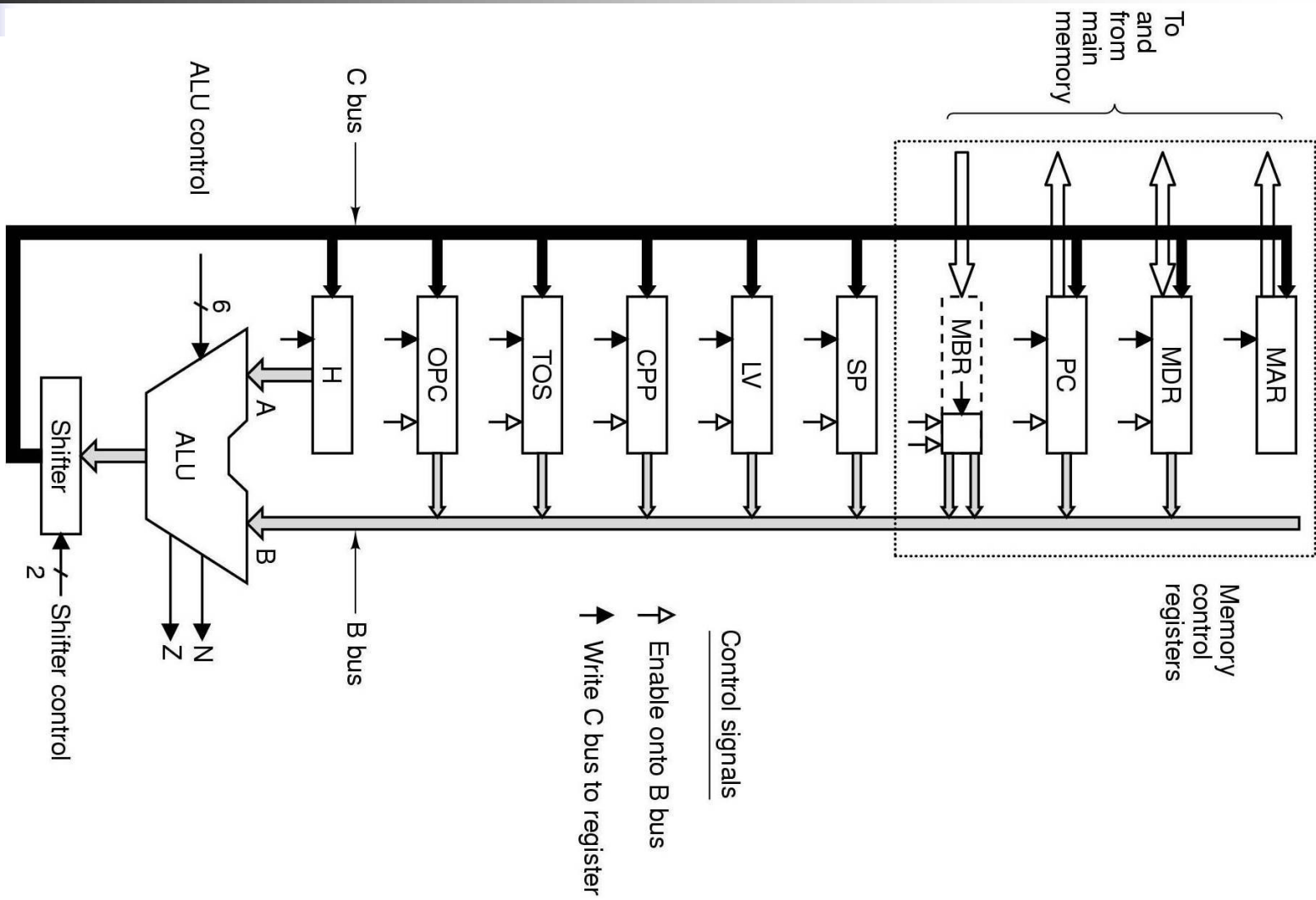


4 Ниво на микроархитектурата (L1)

- Ќе биде разгледувано на примерот на **IJVM** – подмножество од **Java Virtual Machine** кое ги содржи само целобројните (integer) инструкции
- IJVM има некои **комплексни** инструкции – повеќето такви архитектури најчесто се имплементирани со **микропрограмирање**
 - микропрограма ги презема, декодира и извршува IJVM инструкциите

4.1.1 Податочна патека





4.1.1 Податочна патека

- **Податочна патека (data path)** – оној дел од процесорот кој ги содржи аритметичко-логичката единица (ALU), нејзините влезови и нејзините излези
- Содржи одреден број (32-битни) **регистри** со соодветни симболички имиња: PC, SP, MDR, ...
- Содржината на најголем дел од регистрите може да биде прочитана преку **магистралата B**
- Излезот од **ALU**, а потоа и од **поместувачот (shifter)**, преку **магистралата C** може да биде запишан во еден или во неколку регистри истовремено



4.1.1 Податочна патека

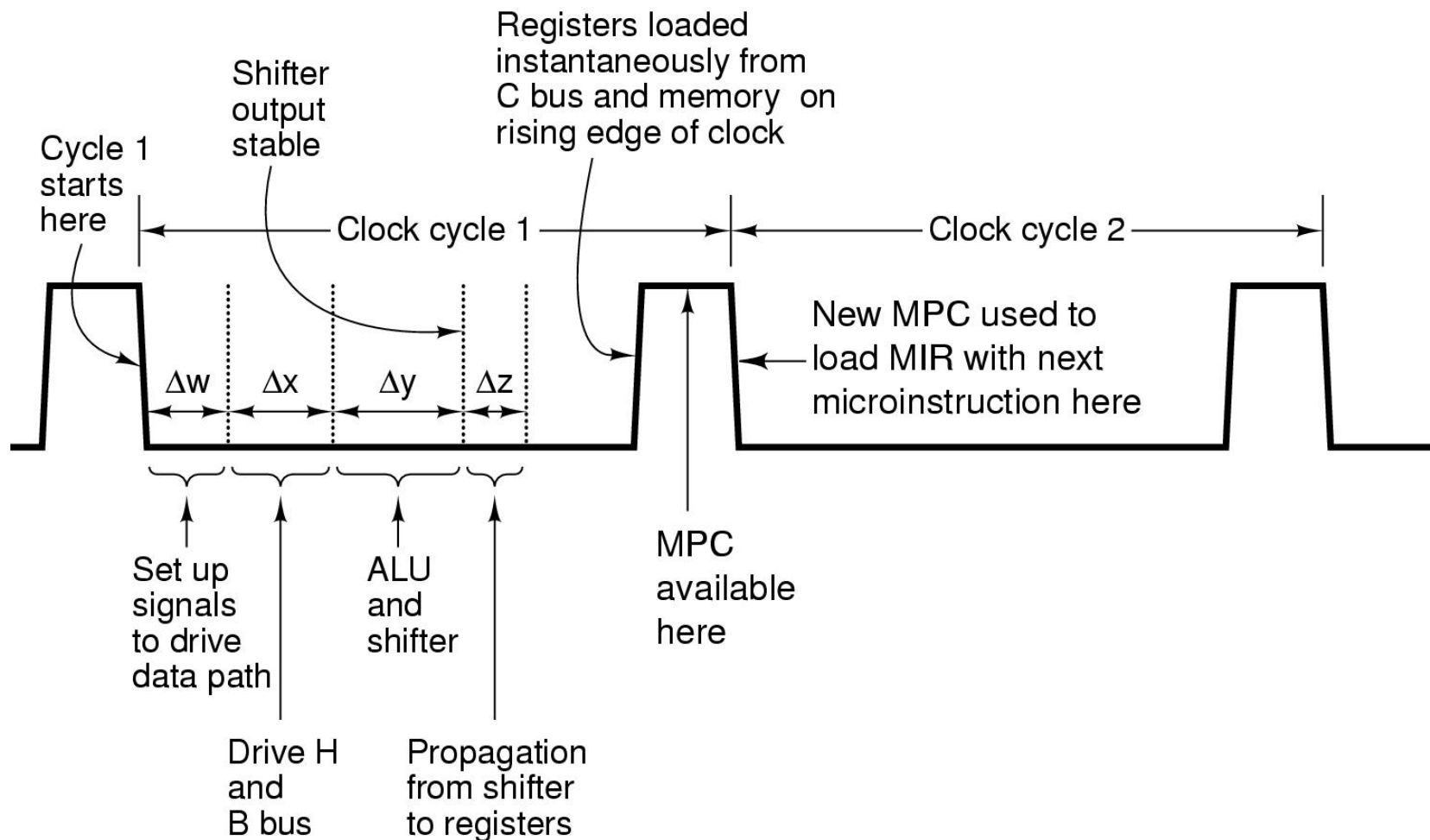
- Функцијата што треба да ја изврши ALU се определува со 6 контролни линии (на сликата претставени заедно):
 - F0 и F1 – ја определуваат аритметичко-логичката операција
 - ENA и ENB – одделно ги овозможуваат (enable) левиот и десниот влез на ALU
 - INVA – го инвертира левиот влез на ALU
 - INC – инкрементирање (додавање 1 на резултатот)
- Два влеза:
 - **Лев влез (A)** – на него е поврзан регистерот **H (holding register)**
 - **Десен влез (B)** – на него е поврзана магистралата B



4.1.1 Податочна патека

- На кој начин може да се наполни регистерот Н со нова содржина?
 - Се избира функција која само го проследува десниот влез (од магистралата В) до излезот од ALU
 - Пример: собирање на двата влеза, при што левиот влез нема да биде овозможен (ENA=0)
- Дополнително, уште две контролни линии можат независно да го контролираат излезот од ALU (на сликата претставени заедно):
 - **SLL8 (Shift Left Logical)** – поместување на содржината за **1 бајт налево** (8-те најмалку значајни битови се пополнуваат со нули)
 - **SRA1 (Shift Right Arithmetic)** – поместување на содржината за **1 бит надесно** (најзначајниот бит останува непроменет)

4.1.1.1 Временски дијаграм на циклусот на податочната патека



4.1.1.1 Временски дијаграм на циклусот на податочната патека

- Почетокот на секој нов циклус на податочната патека се означува со **краток импулс**
- Циклусот е поделен на четири **суб-циклуси**:
 - Поставување на контролните сигнали (Δw – започнува на опаднувачкиот раб (falling edge) од импулсот)
 - Читање на содржината на регистрите на магистралата B (Δx)
 - Извршување на операцијата на ALU и поместувачот (Δy)
 - Проследување на резултатите вдоль магистралата C, назад кон регистрите (Δz)
- Резултатите се запишуваат во регистрите на растечкиот раб (rising edge) од следниот импулс
- Суб-циклусите се **имплицитни** – не постојат други импулси или експлицитни сигнали кои на ALU и соопштуваат **кога** да ја изврши операцијата

4.1.1.2 Комуникација со меморијата

- Две можности за комуникација:
 - 32-битна мемориска порта, контролирана со два регистри:
 - **MAR (Memory Address Register)** – мемориски адресен регистер
 - **MDR (Memory Data Register)** – мемориски податочен регистер
 - 8-битна мемориска порта, контролирана со еден регистер:
 - **PC (Program Counter)** – програмски бројач, кој вчитува 1 бајт од меморијата и го сместува во 8-те најмалку значајни битови во регистерот **MBR**
- За секој од регистрите постојат по два контролни сигнали:
 - Сигнал за овозможување на излезот од регистерот **кон** магистралата В (означен со празна стрелка)
 - Сигнал за запишување во регистерот (полнење на регистерот со содржина од магистралата С)



4.1.1.2 Комуникација со меморијата

- MAR содржи адреси на мемориски зборови
 - Пример: ако содржината на MAR е 2, од меморијата ќе бидат прочитани бајтовите 8-11 (зборот со реден број 2) и ќе бидат запишани во MDR
- PC содржи адреси на бајтови
 - Пример: ако содржината на PC е 2, од меморијата ќе биде прочитан бајтот со реден број 2 и ќе биде запишан во 8-те најмалку значајни битови од MBR
- Разликата во функционалноста е неопходна бидејќи MAR и PC ќе бидат искористени за обраќање кон два различни дела од меморијата (податоци и инструкции?)



4.1.1.2 Комуникација со меморијата

- Содржината на 8-битниот регистер MBR може да биде проследена како 32-битна вредност на магистралата B на два начина:
 - Вредност без знак (unsigned; 0-255) – првите 24 бита се нули
 - Вредност со знак (signed; -128...+127) – битот за знак од MBR (најлевиот бит) се пресликува во сите претходни позиции (sign extension), така што првите 24 бита ќе бидат или сите 0, или сите 1
- Изборот за тоа дали вредноста на MBR ќе биде проследена со или без знак се прави со еден од двата контролни сигнали (прикажани со празни стрелки)



4.1.2 Микроинструкции

- 29 сигнали за контрола на податочната патека:
 - 9 сигнали за контрола на запишувањето на податоци од магистралата C во регистрите
 - 9 контролни сигнали за овозможување на излезите од регистрите кон магистралата B (кон влезот на ALU)
 - 8 сигнали за контрола на функциите на ALU и поместувачот
 - 2 сигнали (не се прикажани) за комуникација со меморијата (читање / запишување) преку MAR/MDR
 - 1 сигнал (не е прикажан) за комуникација со меморија (преземање) преку PC/MBR
- Забелешка: доколку се активира сигнал за читање на податоци од меморијата, комуникацијата започнува на крајот од циклусот (k) на податочната патека (откако MAR ќе биде наполнет), а податоците ќе бидат достапни во MDR или MBR на крајот на **следниот** циклус (k+1) и можат да бидат искористени дури во циклусот кој следува **по него** (k+2) или подоцна

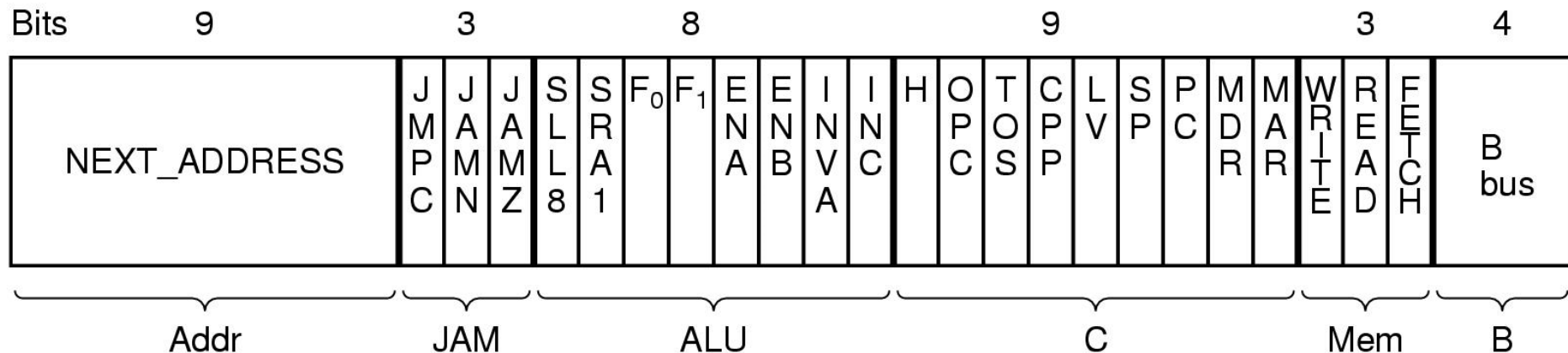


4.1.2 Микроинструкции

- Бидејќи никогаш не е пожелно истовремено проследување на содржината на повеќе од еден регистер на магистралата B, 9-те контролни сигнали за овозможување на излезите можат да се кодираат со 4 бита ($29 \rightarrow 24!$)
- Потребно е, исто така, да се знае **ШТО** треба да се прави во следниот циклус – две дополнителни полиња:
 - **NEXT_ADDRESS** – ја содржи адресата на можната следна микроинструкција
 - **JAM** – определува **како** треба да се избере следната микроинструкција

4.1.2 Микроинструкции

■ Формат на микроинструкциите



B bus registers

0 = MDR	5 = LV
1 = PC	6 = CPP
2 = MBR	7 = TOS
3 = MBRU	8 = OPC
4 = SP	9-15 none



4.1.2 Микроинструкции

- Некои комбинации од ALU сигнали
 - Аритметичко-логичката единица може да пресметува една од следните четири функции:

F_0	F_1	ENA	ENB	INVA	INC	Function
0	0	1	1	0	0	A AND B
0	1	1	1	0	0	A OR B
1	0	1	1	0	0	\bar{B}
1	1	1	1	0	0	A + B

- Како ќе се пресмета $B - A$ (можеби како $B + (-A)$)...

1	1	1	1	1	1	$B - A$
---	---	---	---	---	---	---------