



COMPUTER ORGANIZATION & ARCHITECTURE

MANVITH PRABHU

211EC228

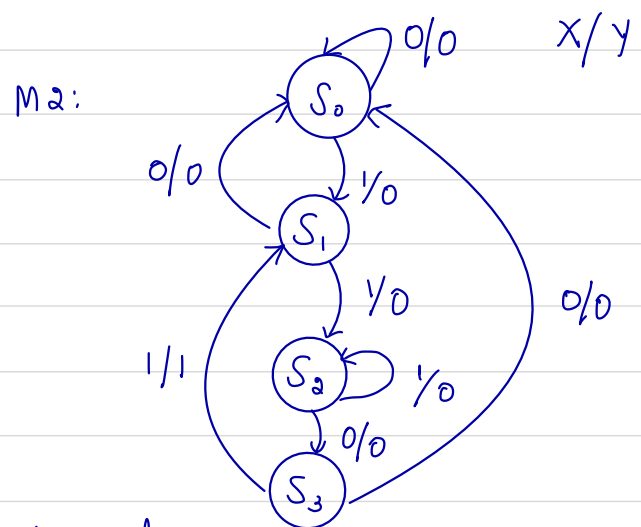
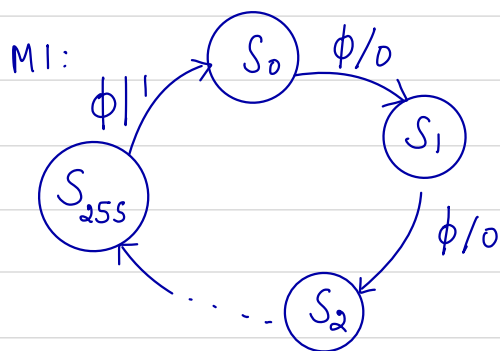


COMPUTER ORGANISATION AND ARCHITECTURE

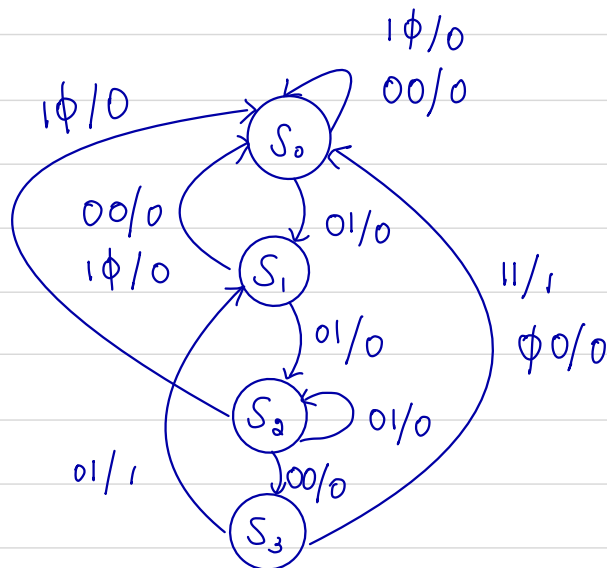
Linked state diagram

Q1> Search for pattern 1101 in a block of 256 bits.

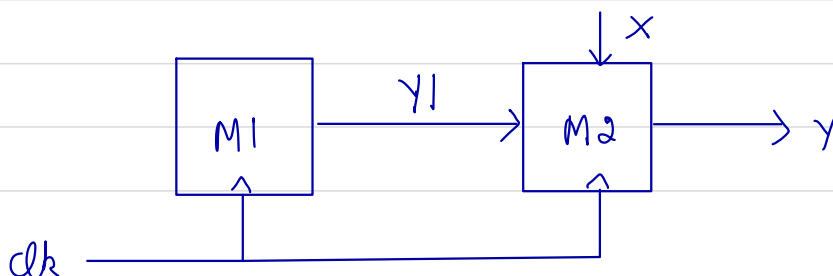
Ans
M1 - module 256 counter
M2 - Pattern recogniser



When M2 takes input from M1:

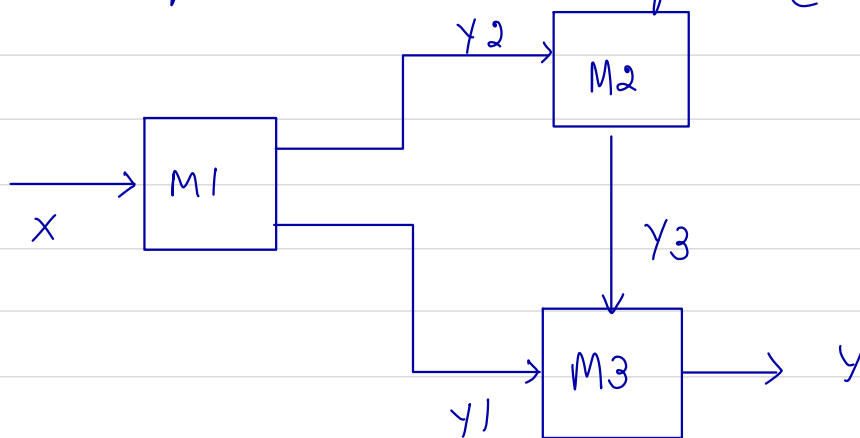


$y1, x/y$



Q2) In a bit stream count the number of @ characters in blocks of 256. Each character is represented by a 8 bit ASCII code.

Ans
 M1 - pattern recognizer
 M2 - 8 bit counter counting module 256
 M3 - 9 bit counter for storing number of occurrence of @ character.



- $Y_1 = 1$ if @ is received (enables M3 counter) if $X_{t-7} \dots X_t = 01000000$ and $t \bmod 8 = 7$
- $Y_2 = 1$ when a character is received if $t \bmod 8 = 7$ (enables M2 counter)

M2: $Y_3 = 1$ if state = S_{255}

M3: $Y =$

$$(-5) \times 3$$

$$\begin{array}{r}
 1011 \\
 \underline{0011} \\
 1111011 \\
 111011 \\
 00000 \\
 \underline{0000} \\
 1110001
 \end{array}
 \Rightarrow (-15)_{10}$$

$$5 \times (-3)$$

$$\begin{array}{r}
 0101 \\
 \underline{1101} \\
 0000101 \\
 0000000 \\
 00101 \\
 \underline{1011} \\
 1110001
 \end{array}
 \Rightarrow (-15)_{10}$$

M
1100
A
0000

Q
01110

① → sign extension

Q recoding: 100(-1)

C	A	Q	
0	0000	1001	Add (-M)
0	0100	1001	↑ (2's complement for -ve)
0	0010	0100	Right shift
0	0001	0010	Right shift
0	0000	1001	Right shift
0	1100	1001	Add M
0	①110	0100	Right shift

∴ n=4 ⇒ 4 right shifts

1110 0100 = -28

Q>

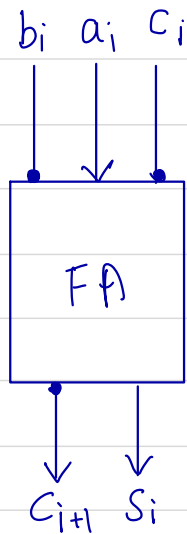
7 ÷ 2 =	0111 / 0010	M = 0111	
Dividend	divisor	Quotient (Q)	Remainder (A)
7	2	3	1
7	-2	-3	1
-7	2	-3	-1
-7	-2	3	-1

$$M = 0010$$

$$-M = 1110$$

	A	Q
Initial	0000	011
Left shift	0000	111_
Sub M	1110	111_
Set $q_0 = 0$	1110	1110
Left shift	1101	110_
Add M	1111	110_
Set $q_0 = 0$	1111	1100
Left shift	1111	100_
Add M	1111	100_
Set $q_0 = 0$	1111	1000
Left shift	1111	000_

NOTE



FA - Full adder
FS - Full subtractor

→ This becomes FS

11-4-23

Q>

Find the dependencies

add \$s3, \$s1, \$s2
and \$s5, \$s3, \$s4
add \$s2, \$s6, \$s3
lw \$s6, 24(\$s3)
sw \$s6, 12(\$s2)

T/F

1> F

2> F

3> T

4> F

5> T

6> F

7> F - 6 stalls

Q1>

Non pipelined: $t_{c_{NP}} = \frac{1}{1.5 \text{ GHz}}$

Instructions: $5 \times t_c$

Pipelined: 20% Branch, 30% MEM instructions, 10% LD-ALU instructions.

30% branch cause 2 cycles stall

5% MEM instruction cause 50 cycles stall

LD-ALU cause 1 cycle

$$t_{cp} = 1 \times 10^{-9} \text{ s}$$

$CPI_{\text{pipeline}} = 1 + \text{mem stalls} + \text{br stalls} + \text{LD-ALU stalls}$

Execution time = $CPI_p \cdot t_{cp}$

$$CPI_p = 1 + 0.05 \times 0.3 \times 50 + 0.2 \times 0.3 \times 2 + 0.1 \times 1$$

$$= 1.97$$

$$\text{Speedup} = \frac{\text{Execution time without pipeline}}{\text{Execution time with pipeline.}}$$

$$= \frac{5 \times (2/3) \times 10^{-9}}{1.97 \times 10^{-9}} = 1.69 \checkmark$$

Dynamic scheduling

1	ADD	R1, R2, R3	} U1
2	ST	R1, (R4 + 50)	
3	ADD	R1, R5, R6	} Register renaming R1 → U3
4	SUB	R7, R1, R8	
5	ST	R1, (R4 + 54)	
6	ADD	R1, R9, R10	

→ renamed → U5

⇒ Register status indicator:

R1	R2	R3	R4	R5	R6	R7	R8	R9	R10
0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	4	0	0	0
6	0	0	0	0	0	4	0	0	0

Instruction fetch

↓

Register status indicator

↓

Exe1	Exe2	Exe3	Exe4	Exe5	Exe6
Empty I1: Ex I1: Ex	Empty I2: W1 I2: W1	Empty I3: W1 I3: Ex	Empty I4: W3 I4: W3	Empty I5: W3 I5: W3	Empty I6: Ex I6: Ex

with →
renaming

W = Wait

Tomasulo's algorithm

L.D F0, 32(R2) Opn V_j V_k Q_j Q_k A
 L.D F2, 44(R3)
 MUL.D F0, F2, F4 $D \rightarrow \text{Double}$
 SUB.D F0, F2, F6
 DIV.D F10, F0, F6
 ADD.D F6, F8, F2

	Busy	Opn	V_j	V_k	Q_j	Q_k	A
Load 1	No Yes	Load	F6	R2			$32 + (R2)$
Load 2	No Yes	Load	F2	R3			$44 + (R3)$
Add 1	No Yes	Sub	-	-	Load 2	Load 1	-
Add 2	No Yes	Add	-	-	Add 1	Load 2	-
Add 3	No	-	-	-	-	-	-
Mul 1	No Yes	Mul	-	F4	Load 2	-	-
Mul 2	No Yes	Div	-	-	Mul 1	Load 1	-
			Independent		dependent		

RS:

F₀ F₂ F₄ F₆ F₈ F₁₀
 Mul 1 Load 2 - Add 2 Add 1 Mul 2
 clocks \rightarrow 1 2 6/40
 Load/store FP adder Fp multiplier

eg 2:

LD F0, 0(R1)
 MUL F4, F0, F2
 SD F4, 0(R1)
 DADDUI R1, R1, -8
 BNE R1, R2, Loop

	Busy		Opn	V _j	V _k	Q _j	Q _k	A
Load 1	No	Yes	Load		R1			(R1)+0
Load 2	No	Yes	Load		R1-8			0+(R1-8)
Add 1	No							
Add 2	No							
Add 3	No							
Mul 1	No	Yes			F ₂	Load 1		
Mul 2	No	Yes			F ₂	Load 2		
Store 1	No	Yes	Store		R1	Mul 1		
Store 2	No	Yes	Store		R1-8	Mul 2		

F0 F2 F4
 Load 1 Mul 1
 Load 2 Mul 4

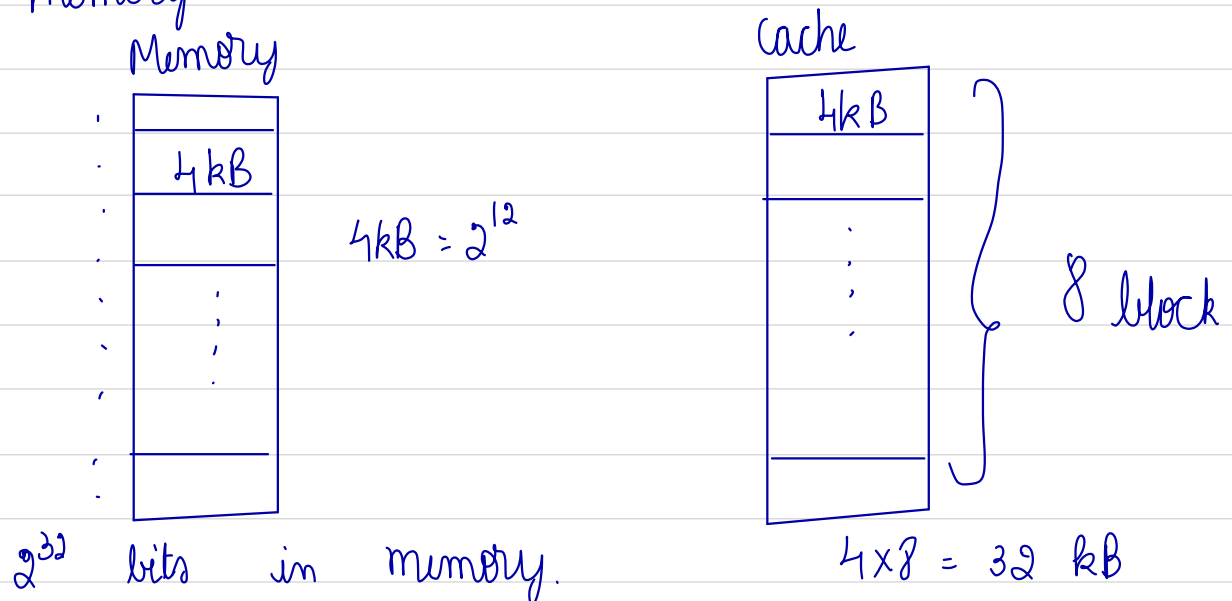
Qs from slides	Instruction	(Cycle number)					
		Issue	Src 1	Src 2	Ex	Write CDB	Commit
	DIV R4, R4, #4	1	RF	IMM	2	10	11
	MUL R2, R6, R12	2	RF	RF	3	7	12
	DIV R1, R1, R2	3	RF	CDB	10	18	19
	ADD R5, R1, R3	4	CDB	RF	19	20	21
	ADDI R7, R2, #4	5	CDB	IMM	8	9	22
	ADD R5, R6, R7	10	RF	ROB	11	12	23
	ADDI R8, R8, #24	13	RF	IMM	14	15	24
	ADD R9, R6, R8	16	RF	ROB	17	19	25
	MUL R5, R5, R19	17	ROB	RF	18	22	26
	ADD R6, R8, R5	20	ROB	CDB	23	24	27

ROB - Reorder Buffer
RF - Register file

IMM - Immediate
CDB - Common data bus

Mapping:

Associative: Placed in any free memory in memory



$$\therefore \text{No. of blocks} = \frac{2^{32}}{2^{12}} = 2^{20}$$

2> 9m set associative, 2 blocks of cache = 1 set

3> Direct

NOTE: Miss: No data in cache to read
Hit: Data already there

Memory:

Q> Tag access - 3ms
Data access - 4.2 ms
hit comparison - 1.3ms
Data transmission - 1ms

Ans Cache hit latency = $\max(3+1.3, 4.2) + 1$
 $= \max(4.3, 4.2) + 1$
 $= 5.3 \text{ ms}$

Q> $(\text{Hit rate})_A = (\text{Hit rate})_B$
 $(\text{Miss rate})_A = (\text{Miss rate})_B$

$(\text{MPKI})_A = 44$ $(\text{MPKI})_B = 35$

If 35% of A is data instruction what is the data access pattern of B if A & B have same cache hit rate.

Ans $\text{MR} = \frac{\text{miss}}{\text{num access}} = \frac{\text{miss instruction}}{\text{num access instruction}}$

$$\frac{44/1000}{1.35} = \frac{35/1000}{1+x}$$

$$x = 0.073$$

$$\therefore \% = 7.3\% //$$

Q> 32 bit processor :
 L1 I cache 8 KB - Direct
 D cache 16 KB - 2 way set assoc
 Block size - 8 words

L2 128 KB 4 way set assoc
 Block size - 16 words

L1, L2 - Inclusive

MM - 4 word address from 22 D D - Decimal

MM 260, 261, 275 D

Non empty blocks in L1 & L2 after exec.

Ans L1 - I cache Blocks : $\frac{8 \times 1024}{8 \times 4} = 256$ blocks

L1 - D cache Blocks : $\frac{16 \times 1024}{8 \times 4} = 512$ blocks = $\frac{512}{2} = 256$ sets

L2 - blocks : $\frac{128 \times 1024}{16 \times 4} = 2048$ blocks = 512 sets

Inclusive : If data is removed from or put into L1, it has to be removed from L2 and vice versa. Data removed is put into victim cache.

Exclusive : Data is put into L1 but not into L2, but if data is removed from L1, it is put into L2.

Non inclusive, non exclusive : Data is put into both, but removed from only one cache.

S = Set

B = Block

L2:

$$B0 = 0 \rightarrow 15$$

$$B1 = 16 \rightarrow 31$$

\therefore 22-25 will go to B1 in L2

S1 in L2

$$\Rightarrow \frac{1}{512}, \text{ remainder} = 1$$

$$260, 261 = B16 \text{ in L2}$$

S16 in L2

$$\Rightarrow \frac{16}{512}, \text{ remainder} = 16$$

$$275 = B17 \text{ in L2}$$

S17 in L2

L1:

22, 23 - B2 I cache

24, 25 - B3 I cache

260, 261 - B32 D cache = S32 cache

275 - B34 D cache = S34 cache

Q> Cache access time = 4 ns

" hit rate = 80%

Main memory access time = 100 ns

" hit rate = 99.5%

Virtual memory access time = 10 ms

Ans

$$4 \times 0.8 + 0.2 \times 100 \times 0.995 + 10 \times 0.2 \times 0.05 \times 10^6$$
$$= 3.2 + 19.9 + 10000$$
$$= 10023 \text{ ns}$$