



COMPUTER ORGANIZATION & ARCHITECTURE

MANVITH PRABHU

211ECA228

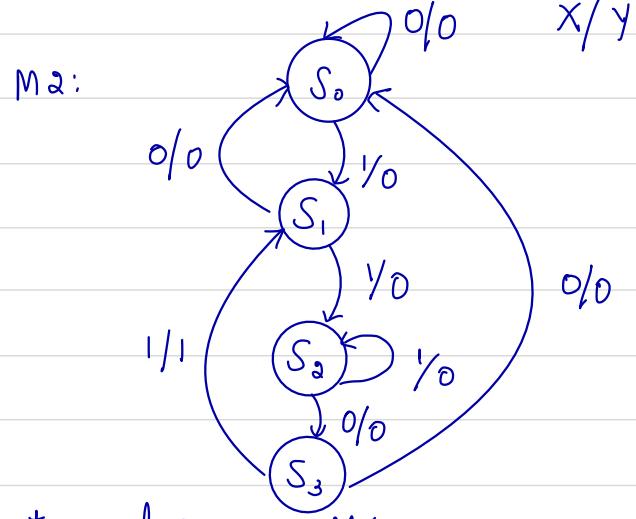
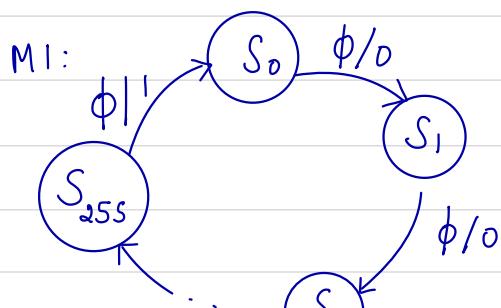


COMPUTER ORGANISATION AND ARCHITECTURE

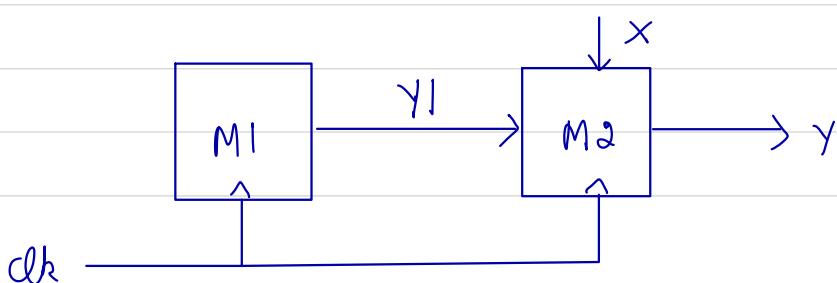
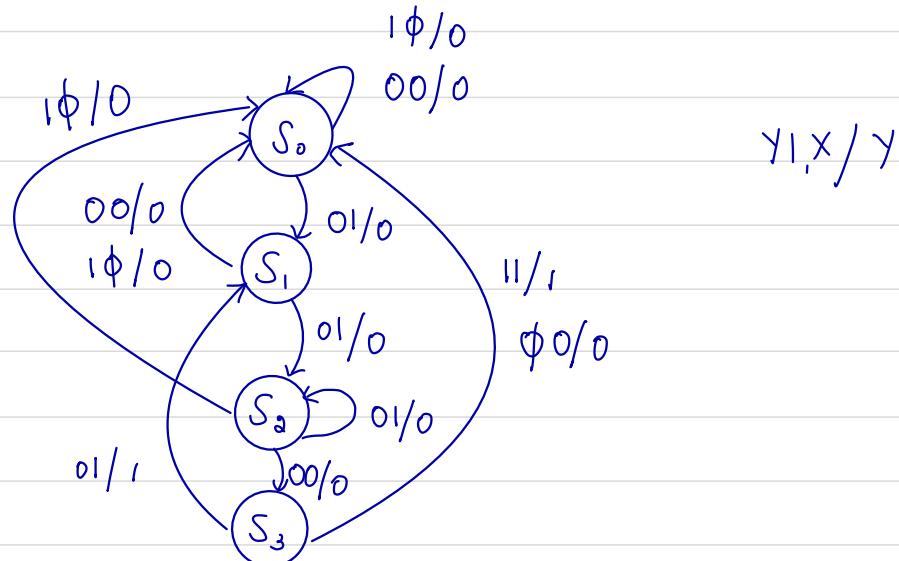
Linked state diagram

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

Ans
 M1 - modulo 256 counter
 M2 - pattern recogniser

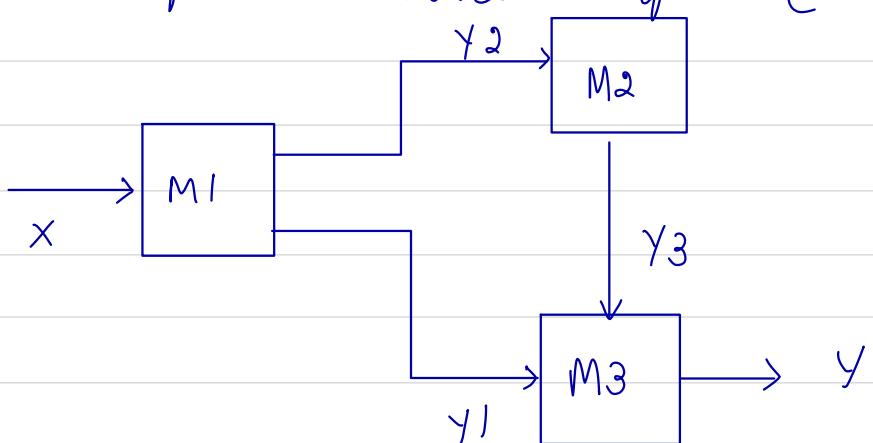


When M2 takes input from M1:



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: $y =$

$(-5) \times 3$

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

$5 \times (-3)$

$$\begin{array}{r} 0101 \\ 1101 \\ \hline 0000101 \\ 0000000 \\ 00101 \\ \hline 10111 \\ \hline 1110001 \end{array} \Rightarrow (-15)_{\text{H}}$$

M		
1100		
A		
0000		Q
		01110

① \rightarrow sign extension

Q recording : 100(-1)

C	A	Q	Add (-M)
0	0000	1001	↑ (2's compliment for -ve)
0	0100	1001	
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

$\therefore n=4 \Rightarrow 4$ right shifts

$$1110 \ 0100 = -28$$

Q> $7 \div 2 = 0111/0010$ $M = 0111$
 Dividend divisor Quotient (Q) Reminder (A)

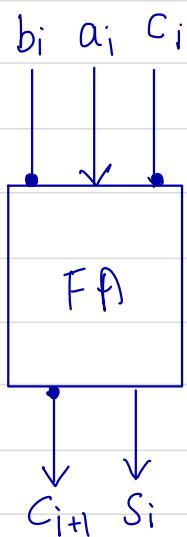
7	2	3	1
7	-2	-3	1
-7	2	-3	-1
-7	-2	3	-1

$$M = 0010$$

$$-M = 1110$$

	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



T/F

1> F

2> F

3> T

4> F

5> T

6> F

7> F - 6 stalls

Q1>

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

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 pipeline = 1 + mem stalls + br stalls + LD-ALU stalls

Execution time = CPI_p \cdot t_{cp}

$$CPI_r = 1 + 0.05 \times 0.3 \times 50 + 0.2 \times 0.3 \times 2 + 0.1 \times 1 \\ = 1.97$$

Speedup = Execution time without pipeline
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)	}	renamed → U5
6	ADD	R1, R9, R10		

⇒ 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

↓

Ex1	Ex2	Ex3	Ex4	Ex5	Ex6
Empty I1: Ex	Empty I2: W1	Empty I3: W1	Empty I4: W3	Empty I5: W3	Empty I6: Ex
with → I1: Ex	I2: W1	I3: Ex	I4: W3	I5: W3	I6: Ex

with →
renaming

W = Wait

Tomarulo's algorithm

L.D	FBusy32(R2)pn	V _j	V _k	Q _j	Q _k	A
L.D	F ₂ , 45(R ₃)					
MUL.D	F ₀ , F ₂ , F ₄					D \Rightarrow Double
SUB.D	F ₀ , F ₂ , F ₆					
DIV.D	F ₁₀ , F ₀ , F ₆					
ADD.D	F ₆ , F ₈ , F ₂					

	Busy	Opn	V _j	V _k	Q _j	Q _k	A
Load 1	No	Yes	Load	F ₆	R ₂		32 + (R ₂)
Load 2	No	Yes	Load	F ₂	R ₃		45 + (R ₃)
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	-	F ₄	Load 2	-
Mul 2	No	Yes	Div	-	-	Mul 1	Load 1
				Independent		dependent	

RS:

F ₀	F ₂	F ₄	F ₆	F ₈	F ₁₀
Mul 1	Load 2	-	Add 2	Add 1	Mul 2
clocks \rightarrow	1	2	3	4	5
Load / store	Fp addr				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	Vj	Vk	Qj	Qk	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		F2	Load1		
Mul 2	No	Yes		F2	Load2		
Store 1	No	Yes	STORE	R1	Mul1		
Store 2	No	Yes	STORE	R1-8	Mul2		

F0 F2 F4
 Load 1 Mul 1
 Load 2 Mul 4

(Cycle number)

Qs
from
slides

Instruction	Issue	Src1	Src2	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:

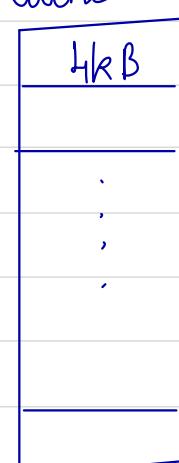
Associative: Placed in any free memory in memory

Memory



$$4KB = 2^{12}$$

Cache



8 block

2^{32} bits in memory.

$$4 \times 8 = 32 \text{ KB}$$

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

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 - 3 ns

Data access - 4.2 ns

hit comparison - 1.3 ns

Data transmission - 1 ns

$$\begin{aligned} \text{Am Cache hit latency} &= \max(3 + 1.3, 4.2) + 1 \\ &= \max(4.3, 4.2) + 1 \\ &= 5.3 \text{ ns} \end{aligned}$$

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

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

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

If 35% of A is data instruction what is the data access pattern of B if A

if B have same cache hit ratio.

$$\begin{aligned} \text{Am} \quad \text{MR} &= \frac{\text{miss}}{\text{num access}} = \frac{\text{miss}}{\text{instruction}} / \frac{\text{num access}}{\text{instruction}} \end{aligned}$$

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

$$x = 0.073$$

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

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 - Icache Blocks : $\frac{8 \times 1024}{8 \times 4} = 256$ blocks

L1 - Dcache 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.

$S = \text{Set}$

$B = \text{Block}$

$$L2: B0 = 0 \rightarrow 15$$

$$B1 = 16 \rightarrow 31$$

$\therefore 22-25 \text{ will go to } B1 \text{ in } L2$

$S1 \text{ in } L2$

$\Rightarrow \frac{1}{S12}, \text{ remainder} = 1$

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

$$S16 \text{ in } L2 \Rightarrow \frac{16}{S12}, \text{ remainder} = 16$$

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

$S17 \text{ in } L2$

$$L1: 22, 23 - B2 \quad I \text{ cache}$$

$$24, 25 - B3 \quad I \text{ cache}$$

$$260, 261 - B32 \quad D \text{ cache} = S32 \text{ cache}$$

$$275 - B34 \quad D \text{ cache} = S34 \text{ cache}$$

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