

EE4218 Computer Architecture
Sem A 2019-20
Assignment I

(a) Write a program to compute

$$X = (A + B \times C) / (D - E \times F)$$

There is a working register, $W \leftarrow$. All the arithmetic instruction has to be conducted through the workig register, The instructions available for use are as follows:

MOV	M	% ($W \leftarrow M$), The data is at memory location M	FI DI FO EI
MOVW	M	% ($M \leftarrow W$)	FI DI EI WO
ADD	M	% ($W \leftarrow W + M$)	FI DI FO EI
SUB	M	% ($W \leftarrow W - M$)	FI DI FO EI
MUL	M	% ($W \leftarrow W \times M$)	FI DI FO EI
DIV	M	% ($W \leftarrow W / M$)	FI DI FO EI

Y is needed for either to store the denominator or nominator of the equation.

			Stage involved
1. MOV	E	% $W \leftarrow [E]$	FI DI FO EI
2. MUL	F	% $W \leftarrow [W] \times [F] = [E] \times [F]$	FI DI FO EI
3. MOVW	X	% $X = E \times F$	FI DI EI WO
4. MOV	D	% $W \leftarrow [D]$	FI DI FO EI
5. SUB	X	% $W = D - E \times F$	FI DI FO EI
6. MOVW	X	% $X \leftarrow [W]$	FI DI EI WO
7. MOV	B	% $W \leftarrow [B]$	FI DI FO EI
8. MUL	C	% $W \leftarrow [W] \times [C] = [B] \times [C]$	FI DI FO EI
9. MOVW	Y	% $Y: B \times C$	FI DI EI WO
10. MOV	A	% $W \leftarrow [A]$	FI DI FO EI
11. ADD	Y	% $W = A + B \times C$	FI DI FO EI
12. DIV	X	% $W \leftarrow W/X$	FI DI FO EI
13. MOVW	X	% $X \leftarrow [W]$	FI DI EI WO

(b) For each of the 6 instructions, show the addressing modes of source operand and destination operand, and the number of memory references.

		Add. Mode	SO	DO	MRef
MOV	M	all direct	M	W	all one
MOVW	M		W	M	
ADD	M		M	W	
SUB	M		M	W	
MUL	M		M	W	
DIV	M		M	W	

- (c) A microprocessor has a pipeline with 5 stages:
- fetch instruction (FI),
 - decode instruction (DI),
 - fetch **ONE** source operand from memory (FO),
 - execute instruction (EI), and
 - write **ONE** destination operand to memory (WO).

Instructions are issued at a rate of one per clock cycle. Assume that each pipeline stage takes one clock cycle. FI, FO and WO cannot occur simultaneously.

Draw the timing diagram showing the pipeline operation in running the program in part (a).

Y is needed for either to store the denominator or nominator of the equation.

			Stage involved
1. MOV	E	% W <- [E]	FI DI FO EI
2. MUL	F	% W <- [W] x [F] = [E] x [F]	FI DI FO EI
3. MOVW	X	% X = E x F	FI DI EI WO
4. MOV	D	% W <- [D]	FI DI FO EI
5. SUB	X	% W = D - E x F	FI DI FO EI
6. MOVW	X	% X <- [W]	FI DI EI WO
7. MOV	B	% W <- [B]	FI DI FO EI
8. MUL	C	% W <- [W] x [C] = [B] x [C]	FI DI FO EI
9. MOVW	Y	% Y: B x C	FI DI EI WO
10. MOV	A	% W <- [A]	FI DI FO EI
11. ADD	Y	% W = A + B x C	FI DI FO EI
12. DIV	X	% W <- W/X	FI DI FO EI
13. MOVW	X	% X <- [W]	FI DI EI WO

Pipeline

	t1	t2	t3	t4	t5	t6	t7	t8	t9	t10	t11	t12	t13	t14	t15	t16	t17
1	FI	DI	FO	EI													
2		FI	DI	FO	EI												
3			FI	FI	FI	DI	EI	WO									
4						FI	DI	FO	FO	EI							
5							FI	DI	FO	FO	EI						
6								FI	FI	FI	FI	DI	EI	WO			
7										-		FI	DI	FO	FO	EI	
8												-	FI	DI	FO	FO	EI

	t1 3	t14	t15	t16	t17	t18	t19	t20	t21	t22	t23	t24	t25	t26	t27	t28	t29
8	FI	DI	FO	FO	EI												
9		FI	DI	EI	WO												
10			FI	DI	FO	EI											
11				FI	FI	FI	DI	FO	EI								
12							FI	DI	FO	EI							
13								FI	FI	FI	DI	EI	WO				

2. Design a set-associative cache for a set size of 4 to accommodate a total of 2048 words and line size of 4. The main memory size is 64K 32 bits. Show the cache structure and how the processor's addresses are interpreted.

For the cache, set size K: 4, line size: 4 words; and 32 bit per word,
no. of line : $2048/4 = 512$, no. of set : $512/4 = 128$

The structure of cache:

S0 : 4 Lines	L0: 4 words . L3: 4 words	Set Size = 128: 2^7 Set Index (SI): 7 bits Line Size: 4 words 2^2 Line Office: 2 bits
.		
S127: 4 lines		

Set Index

For the main memory, 64K 32 bits, 32 bit per word,

Physical address: $64K = 2^6 \times 2^{10}$: 16 bits + 2 bits for 32bit data = 18 bits

Block size = Line size = $4 = 2^2$

Block offset (BO) : 2 bits (to identify which word within a line)

Physical Address 18 bits			
17	9	8	2 1 0
TAG BIT		SI	BO
9 bits		7 bits	2 bits

3. The following table shows the execution times, in seconds, for three different benchmark programs on three machines.

Benchmark	Processor		
	X	Y	Z
1	38	51	122
2	5805	4218	3309
3	407	386	233

Compute the overall SPEC rating for each system using Y as the reference machine.

Benchmark	Processor		
	X	Y	Z
SPEC rating 1	1.34	1	0.42
SPEC rating 2	0.73	1	1.27
SPEC rating 3	0.95	1	1.66
Overall SPEC rating	0.98	1	0.96

May used the Gmean for the Overall SPEC rating

4. A non-pipelined processor has a clock rate of 2.5 GHz and an average CPI (cycles per instruction) of 4. An upgrade to the processor introduces a five-stage pipeline. However, due to internal pipeline delays, such as latch delay, the clock rate of the new processor must be reduced to 2GHz.

a. What is the speedup achieved for a typical program?

For clock rate 2.5GHz, the clock cycling time = $1/f = 4 \times 10^{-10}$ sec or 400psec

For clock rate 2 GHz, the clock cycling time = 5×10^{-10} sec or 500psec

For non-pipelined processor, it takes 4 clock cycle to finish an instruction.

Thus, it takes 1600n psec to finish n instruction.

For a pipelined processor of k : stage, where k is 5, it takes 5 clock cycle to finish the first instruction, and takes one clock cycle to complete an instruction other than the first. Thus, for n instructions, it takes, $[5 + (n-1)]$ 500psec to finish n instructions.

The speed up is: $1600 n / \{ [5 + (n - 1)] 500 \}$.

If n is large, the speed up is approximated as $1600/500 \approx$ a factor of 3.

b. What is the MIPS rate for each processor?

For the first processor, each instruction takes 4 clock cycle, so the MIPS rate is $2500 \text{ MHz} / 4 = 625 \text{ MIPS}$. For the second processor, instructions are completed at the rate of one per clock cycle, so that the MIPS rate is 2000 MIPS.