Second Tutorial

Question 1

- From the information given in Zargham fig. 1.9 compute:
  - Instruction troughput
  - Maximum memory access time
  - Utilisation of the floating-point unit
  - Maximum rate of floting –point instructions (MFLOPS)

Second Tutorial

Question 1 - Answer – [Zargham, Figure 1.9]

Instruction Class	Instruction Frequency %(IF)	Cycles per Instruction (CPI)	Weighted CPI (IF×CPI)
Load and store	30.4	1.5	0.456
Integer add and substract	10.0	1	0.1
Integer multiply and divide	3.8	10	0.38
Floating-point add and subtract	9.5	7	0.665
Floating-point multiply and divide	6.5	15	0.975
Logical	3.0	1	0.03
Branch	20.0	1.5	0.3
Compare, shift system	16.8	2	0.336

Cycles per average insruction = 3.242

Executon time of an average instruction
(for when τ = 10) = 3.242 * 10 = 32.42 nanoseconds

MIPS =(1/32.42) * 1000 = 30.845

Second Tutorial

Question 1 - Answer – 1.1, 1.2 and 1.3

- Instruction troughput

Average instruction throughput $= 1/32.4 \times 10^{-9}$ IPS
$= 30.8 \times 10^{6}$ IPS
$= 30.8 \times$ MIP

- Maximum memory access time

Load requires 1.5 cycles therefore the max memory access time $= 1.5 \times 10$ ns $= 15$ ns

- Utilisation of the floating-point unit

$$= \frac{\text{F.P. Cycles}}{\text{Av. Instruction Cycles}}$$

$$= \frac{0.665 + 0.975}{3.242} = 0.51$$

Second Tutorial

Question 1 - Answer – 1.4

- Maximum rate of floting –point instructions (MFLOPS)

The fastest floating-point instruction is the add/sub. requiring 7 cycles or 70 ns:

Max floating-point rate $= 1/70 \times 10^{-9}$ FLOPS
$= 14.3 \times 10^{6}$
$= 14.3 \times$ MFLOPS

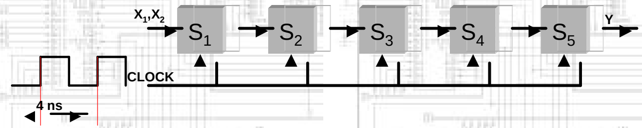
Second Tutorial

Question 2

- A MISD of five statges and with a clock period of 4ns is processing an input vector x.
  - What is its response time, i.e. time to complete processing the first element of x?
  - Compute the throughput and utilisation for vectors of length n = 1, 10, 100

Second Tutorial

Question 2 – Answer – Five stage MISD

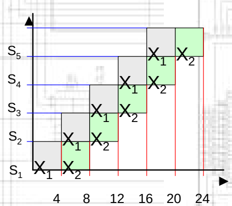


- Each datum will spend 4 ns in each stage, so X will arrive at Y after $5 \times 4 = 20$ ns = Resopnse time

Second Tutorial

Question 2 – Answer – Throughput and Utilisation

- We can show the progress of X_i through the MISD as follows:



Second Tutorial

Question 2 – Answer – Throughput and Utilisation - N=1

- N=1

1 task in 20 ns so

$$\text{Throughput} = \frac{1}{20 \times 10^{-9}} = 50 \text{ MTASKS}$$

$$\text{Utilisation} = \frac{\text{Working Time}}{\text{Total Time}} = \frac{5 \times 1 \times 4}{5 \times 5 \times 4} = 0.2$$

Second Tutorial

Question 2 – Answer – Throughput and Utilisation - N=10

- N=10

10 task in $20 + 9 \times 4 = 56$ ns

$$\text{Throughput} = \frac{10}{56 \times 10^{-9}} = 178.6 \text{ MTASKS}$$

$$\text{Utilisation} = \frac{\text{Working Time}}{\text{Total Time}} = \frac{5 \times 10 \times 4}{5 \times 14 \times 4} = 0.7$$

Second Tutorial

Question 2 – Answer – Throughput and Utilisation - N=100

- N=100

10 task in $20 + 99 \times 4 = 416$ ns

$$\text{Throughput} = \frac{100}{416 \times 10^{-9}} = 240.4 \text{ MTASKS}$$

$$\text{Utilisation} = \frac{\text{Working Time}}{\text{Total Time}} = \frac{5 \times 100 \times 4}{5 \times 104 \times 4} = 0.96$$

- Conclusions
  - As $n \rightarrow \infty$ Throughput $\rightarrow$ 250 Mtasks
    Utilisation $\rightarrow$ 1.0
  - To work at 50% utilisation MISD need n=length

Second Tutorial

Question 3

- A memory is organised as 4M words each of 8 bytes, i.e. 32 Mbytes in all, and to access one word takes 25ns, whereas to sequentially access 1Kwords requires 665 ns. Compute the:
  - Memory access time
  - Memory bandwidth

Second Tutorial

Question 3 – Answer - Memory Access Time and Bandwidth

- Access time to memory involves two compoents

t_L Latency time to decode address and read/write one word

t_D Delivery time per word

$25 \text{ ns} = t_L + 1 \times t_D$

$665 \text{ ns} = t_L + 2^{10} \times t_D$

$\therefore$ $640 \text{ ns} = (2^{10} - 1) \times t_D$

$\therefore$ $t_D = 0.63$ ns/word

$t_L = 25 – 0.63 = 24.37$ ns

$$B_{peak} = \frac{1}{0.63 \times 10^{-9}} = 1.56 \text{ GWORDS/sec}$$

$$B_{wbw} = \frac{1}{25 \times 10^{-9}} = 40 \text{ MWORDS/sec}$$

Second Tutorial

Question 4

- Each processor element (PE) of a SIMD with 2^{16} PEs can complete a floating point addition in 125 ns.
  - What is its maximum troughput in terms of MFLOPS?

Second Tutorial

Question 4 - Answer

- One PE does a floating point add in 125ns

$\therefore$ One PE has throughput $= \dfrac{1}{125 \times 10^{-9}} = 8\text{MFLOPS}$

$\therefore$ 2^{16} PEs have Maximum Throughput

$$MT = 2^{16} \times 8 = 2^{19}$$
$$= 5.2 \times 10^{5}\ \text{MFLOP}$$
$$= 520\ \text{GFLOP}$$

Second Tutorial

Question 5

- The SIMD of question 4 executes the following code on vectors x,y,z of length n:

```
FOR i =1 to n
    IF (y_i ≥ 0) THEN  z_i ← x_i + y_i
                ELSE  z_i ← x_i – y_i
END
```

- If each PE takes 15 ns to complete the sign test on y_i
  - How long will this loop body take to execute?
  - What will be the average throughput and utilisation if n = 2^{16}, 2^{15}, 2^{17}.

Second Tutorial

Question 5 – Answer –

How long will this loop body take to execute?

For each participating PE we have the schedule

Action	Time [ns]	#PEs
IF	15	n
THEN ADD	125	k pass test
ELSE SUB	+125	n-k fail test
LOOP TIME	265	for one f.p

$$\text{Utilisation} = \frac{\text{BUSY TIME}}{\text{TOTAL TIME}} = \frac{125\text{ns} + 15\text{ns}}{265\text{ns}} = 0.53$$

f.p. throughput $= 1 / (265 \times 10^{-9}) = 3.8$ MFLOPS

Second Tutorial

Question 5 – Answer –
What will be the average throughput and utilisation if n = 2^{16}, 2^{15}, 2^{17}

$n = 2^{16}$ ∴ every PE participates

Throughput = $2^{16} \times 3.8$ = 249037 MFLOPS

$$\text{Utilisation} = \frac{\text{BUSY TIME}}{\text{TOTAL TIME}} = \frac{2^{16} \times (125\text{ns} + 15\text{ns})}{2^{16} \times 265\text{ns}} = 0.53$$

$n = 2^{15}$ ∴ 50% PEs idle

Throughput = $2^{15} \times 3.8$ = 124518 MFLOPS

$$\text{Utilisation} = \frac{\text{BUSY TIME}}{\text{TOTAL TIME}} = \frac{2^{15} \times (125\text{ns} + 15\text{ns})}{2^{16} \times 265\text{ns}} = 0.26$$

$n = 2^{17}$ ∴ Each PE has to perform two executions of the loop body.

Throughput = $2^{16} \times 3.8$ MFLOPS

$$\text{Utilisation} = \frac{\text{BUSY TIME}}{\text{TOTAL TIME}} = \frac{2^{16} \times (125\text{ns} + 15\text{ns})}{2^{16} \times 265\text{ns}} = 0.53$$

Second Tutorial

Question 6

- A MIMD multicomputer is constructed from SISD processors with the instructions charcteristics given in Zargham fig. 1.9.
- The task of finding all roots to f(x)=0 is distributed over four PEs and to locate a root requires on average 500 instruction cycles.
- If the actual distribution of roots in f(x) is such that one PE finds seven, one four, one three and one none, compute:
  - How long the MIMD took to locate the fourteen roots?
  - The utilisation of the four PEs
  - How long the task would have taken on one PE?
  - The average throughput
  - The peak throughput

Second Tutorial

Question 6 – Answer – [Zargham, Figure 1.9]

Instruction Class	Instruction Frequency %(IF)	Cycles per Instruction (CPI)	Weighted CPI (IF×CPI)
Load and store	30.4	1.5	0.456
Integer add and substract	10.0	1	0.1
Integer multiply and divide	3.8	10	0.38
Floating-point add and subtract	9.5	7	0.665
Floating-point multiply and divide	6.5	15	0.975
Logical	3.0	1	0.03
Branch	20.0	1.5	0.3
Compare, shift system	16.8	2	0.336

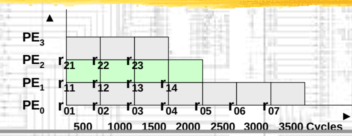
Cycles per average insruction = 3.242

Executon time of an average instruction
(for when τ = 10) = 3.242 * 10 = 32.42 nanoseconds

MIPS =(1/32.42) * 1000 = 30.845

Second Tutorial

Question 6 – Answer - Graphically



Time to locate 14 roots = 7× 500 = 3500 cycles

$$\text{Utilisation} = \frac{\text{BUSY TIME}}{\text{TOTAL TIME}} = \frac{14 \times 500}{4 \times 7 \times 500} = 0.5$$

Time on one PE = 14 × 500 = 7000 cycles

Average troughput = 14 roots / 3500cycles = 1 root / 250 cycles

Peak troughput = 3 roots / 500cycles = 1 root / 167 cycles

Second Tutorial

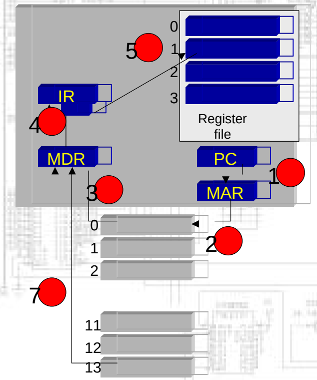
Question 7

- Show the flow of instructions and data on fig. 2.7 for:
  - An INCR instruction (Rd ← Rd + 1)
  - A BRZ Mem_Addr. Instruction (Z:PC ← Mem_addr)

Second Tutorial

Question 7- Answer

Rd ← Rd + 1



0 PC ← 0

1 MAR ← PC

2 ADDR ← MAR

3 MDR ← M[ADDR], PC ← PC + 1

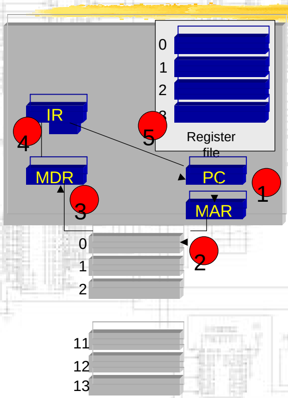
4 IR ← MDR

5 R1 ← R1 +1

Second Tutorial

Question 7- Answer

Z:PC ← Mem_addr



0 PC ← 0

1 MAR ← PC

2 ADDR ← MAR

3 MDR ← M[ADDR], PC ← PC + 1

4 IR ← MDR

5 Z:PC ← IR(3 downto 0)

Second Tutorial

Question 8

- State which of the four instructions given in fig. 2.6 you think will be the fasted and which the slowest to execute, and why.

Second Tutorial

Question 8 – Answer - [Zargham, Figure 2.6]

Opcode	Operand	Operand	Operand

$R_d \leftarrow$ MEMORY[ADDRESS]

LOAD	00	R_d	Memory Address
	7 6	5 4	3 0

MEMORY[ADDRESS] $\leftarrow R_S$

STORE	01	R_S	Memory Address
	7 6	5 4	3 0

$R_d \leftarrow R_{S1} + R_{S2}$

ADDR	10	R_d	R_{s1}	R_{s2}
	7 6	5 4	3 2	1 0

$R_d \leftarrow R_d$ + MEMORY[ADDRESS]

ADDM	11	R_d	Memory Address
	7 6	5 4	3 0

Second Tutorial

Question 8 – Answer

- The fastet will be ADDR since it invalves no meory access.
- The slowest will be ADDM since it involves a memory access followed by an addition opteration.

Second Tutorial

Question 9

- Convert the following microcode, where A, B, ..., H are control signals, NA is the Next Address, and LOC is the location in the CROM at which these words are stored, into efficient vertical microcode:

LOC	A	B	C	D	E	F	G	H	NA
0000	1	0	1	0	0	0	0	0	0001
0001	0	0	0	1	0	0	0	0	0010
0010	0	0	0	0	0	1	0	0	0101
0011	0	0	1	0	0	0	0	0	0011
0100	0	0	0	0	1	0	0	0	0000
0101	0	0	0	0	0	0	0	1	0110
0110	0	1	1	0	0	0	0	0	0000
0111	0	0	0	0	0	0	1	0	1000
1000	0	0	1	0	0	0	0	0	1001
1001	0	0	0	0	0	1	0	0	0000

Second Tutorial

Question 9 - Answer

- We observe that, except for 6 the other control signals are one-out-of-seven. Hence assign:
  - A=1, B=2, D=3, E=4, F=5, G=6, H=7
- And generate the 3-bit SEL address for a 3-to-8 decoder, as well as C and NA

LOC	A	B	C	D	E	F	G	H	NA
0000	1	0	1	0	0	0	0	0	0001
0001	0	0	0	1	0	0	0	0	0010
0010	0	0	0	0	0	1	0	0	0101
0011	0	0	1	0	0	0	0	0	0011
0100	0	0	0	0	1	0	0	0	0000
0101	0	0	0	0	0	0	0	1	0110
0110	0	1	1	0	0	0	0	0	0000
0111	0	0	0	0	0	0	1	0	1000
1000	0	0	1	0	0	0	0	0	1001
1001	0	0	0	0	0	1	0	0	0000

LOC	C	SEL	NA
0000	1	001	0001
0001	0	011	0010
0010	0	101	0101
0011	1	000	0011
0100	0	100	0000
0101	0	111	0110
0110	1	010	0000
0111	0	100	1000
1000	1	000	1001
1001	0	101	0000