NAME: ADITYA DHANARAJ KUNDU

COMPUTER ORGANISATION & ARCHITECTURE

YCS 4001

LAB

10/Feb/2023

SPIM → Span Over Instant Messages

The newest version of SPIM is called QTSPIM. It runs on MS Windows, Macintosh, Linux.

- How to install SPIM (MIPS → Million Instructions Per Second) simulator?

⇒ Go to Google and directly install it. It contains console and...

① Run the program from beginning to end? (via the play/continue button)

② Step through the program one line at a time. (F10) (Run = F5) ↩

③ Run the program until you reach a breakpoint, it is set by right click.

④ There are 3 primary sections in QTSPIM.

(i) Register Panel (ii) Memory Panel (iii) Message Panel

The Register Panel shows the content of all the MIPS registers, there are two tabs in this panel. → ① floating point register ② integer register.

The integer register include the general purpose registers ($R_1 - R_{31}$) along with special purpose register

Memory Panel: The memory panel has 2 tabs data and text. The text tab shows the content of the program memory space from left to right. This includes, ① The memory address of an instruction in hexadecimal.

② The content of that memory address in hexadecimal. in binary form. This is the actual MIPS instruction that the processor runs.

③ The human readable assembly language instruction using the hardware registered number.

(iv) The assembly language program using symbolic register. Names and memory address symbols.

Place (05) in register B.

```
MVI B, 05
HLT
```

2000	06
2001	0C
2002	96

A	
B	C
D	E
H	L

Pipeline Problem

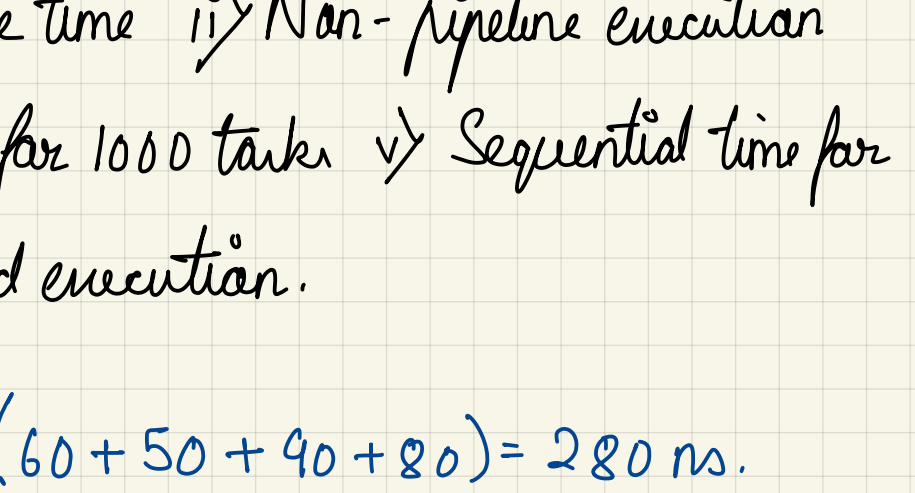
Q.(1) Consider a pipeline having 4 stages with duration 60, 50, 90, 80 ns. Given latch delay is 10 ns. Calculate i) pipeline cycle time ii) Non-pipeline execution time iii) Speed up ratio iv) pipeline tasks for 1000 tasks v) Sequential time for 1000 tasks vi) through put for pipelined execution.

Soln: Non-pipelined execution time = (60 + 50 + 90 + 80) = 280 ns.

Pipeline cycle time = Maximum delay due to any stage + delay due to register (latch delay)

= Max {60, 50, 90, 80} + 10 ns = 90 + 10 = 100 ns

$$\text{Speedup} = \frac{\text{Non-pipeline execution time}}{\text{pipeline execution time}} = \frac{280}{100} = 2.8$$



pipeline time for 1000 tasks:

pipeline cycle count for N tasks (m no. of stage)

= m + (N − 1) × 1

= m + N − 1

= 5 + 4 − 1 = 8

• Multiply two numbers using Booth algorithm.

$(7)_{10} \times (3)_{10} = (21)_{10}$

Sequence Counter = no. of bits = no of cycles.

∴ here no of bits = 4

∴ SC = 4

Ans:

$M \rightarrow (7)_{10} = (0111)_2$ { Multiplicand }

$Q \rightarrow (3)_{10} = (0011)_2$ { Multiplier }

$-M$ = in 2's Complement = 0111

1's Comp = 1000

+1

1001 → $-M$

	AC	Q	Q_{-1}	Operation
①	0000	0011	0	① AC = AC − M = 0000 + 1001 = 1001
	1001	0011	0	
	1100	1001	1	② Arithmetic right shift
②	1110	0100	1	for 11/00, A.R.S Arithmetic Right Shift
③	0101	0100	1	① AC = AC + M = 1110 + 0111 = 0101
	0010	1010	0	② ARS
④	0001	0101	0	ARS

Answer,

16 8 4 2 1

$(00010101)_2 = (21)_{10}$ ∴ $(0111)_2 \times (0011)_2 = (00010101)_2$

$(7)_{10} \times (3)_{10} = (21)_{10}$ Ans

• Multiply -7 and +3 with Booth algorithm.

$M = (-7) = 1001$

$-M = (7) = 0111$

$Q = (3) = 0011$

0111

1's → 1000

+1

2's → 1001

	AC	Q	Q_{-1}	Operation	
1	0000	0011	0	Initialize	
	0111	0011	0	AC = AC − M	0000 + 0111 = 0111
	0011	1001	1	ARS	
2	0001	1100	1	ARS	
3	1010	1100	1	AC = AC + M	0001 + 1001 = 1010
	1101	0110	0	ARS	
4	1110	1011	0	ARS	

→ Answer in 2's Complement,

1110 1011

1's → 0001 0100

+1

0001 0101 = $(00010101)_{16\,8\,4\,2\,1} = (-21)_{10}$ Ans

• Multiply -7×-3 in Booth algorithm.

Ans: $M = (-7) = 1001$

$-M = (7) = 0111$

$Q = (-3) = 1101$

0111
1's 1000
+1
1001 (2's)

0011
1's 1100
+1
1101

	AC	Q	Q_{-1}	Operation
①	0000	1101	0	initialize
	0111	1101	0	AC = AC − M (0001 + 0111 = 0111)
	0011	1110	1	ARS
②	1100	1110	1	AC = AC + M (0011 + 1001 = 1100)
	1110	0111	0	ARS
③	0101	0111	0	AC = AC − M (1110 + 0111 = 0101)
	0010	1011	1	ARS
④	0001	0101	1	ARS

↳ Answer

$(00010101)_2 = (21)_{10}$ (16 8 4 2 1)

Ans

◎ Restoring data algorithm (Slow algorithm) :-

17 ÷ 3, 17 is dividend
3 is divisor

here C & AC are considered together.

Q → (17) = 10001, M → (3) = 000011 n+1 bits to match extra borrow bit.

−M = (−3) = 111100 (→1) = 111101

Sequence Counter = 5 as {n = 5}

C	AC	Q	Operation
0	00000	10001	initialize
0	00001	0001□	L Shift
1	11110	0001 0	C.AC = C.AC − M
0	00001	00010	

As C = 1 so,

Restore
C.AC = C.AC + M

```
  000001
+ 111101
--------
  111110

  111110
+ 000011
--------
  000001
```

C	AC	Q	Operation
0	00010	0010□	L Shift
1	11111	0010 0	AC = AC − M
0	00010	00100	AC = AC + M

```
  000010
  111101
--------
  111111

  111111
+ 000011
--------
  000010
```

C	AC	Q	Operation
0	00100	0100□	L shift.
0	00001	0100 1	AC = AC − M

```
  000100
  111101
--------
  000001
```

C	AC	Q	Operation
0	00010	1001□	LS
1	11111	1001 0	AC = AC − M
0	00010	10010	AC = AC + M

```
  000010
  111101
--------
  111111

  111111
+ 000011
--------
  000010
```

⑤

0	00101	0010	
0	00010	0010	1
0	00010	00101	

LS

AC = AC - M

$$\begin{array}{r} 000101 \\ 111101 \\ \hline 000010 \end{array}$$

Quotient $(00101)_2 = (5)_{10}$

Remainder $(00010)_2 = (2)_{10}$