Name : Shubham Bawner Pages : 9

Roll no : 20118100

Branch : IT

Date : 25 - 5 - 22

Sub : CO End Semister exam.

Q1

A] Selection inputs :

Given : bus is constructed with multiplexcers

∴ we have 16 line buss system.

also, 32 ragisters = 2^5

∴ 5 selection inputs in each multiplier.

ans : 5

B) Size ~~of~~ of multiplexcer needed = 32 X 1

c] No. of multiplexcer required =

= No. of lines in buss system

= 16 : ans

D) Since no. of 3 state buffers required is equal to No. of ragister,

∴ 32 3 state buffers required.

Q3 DMA Controller in Computer Architecture

A hardware device that allows I/O devices to directly access memory with least participation of processor.

Requirments:

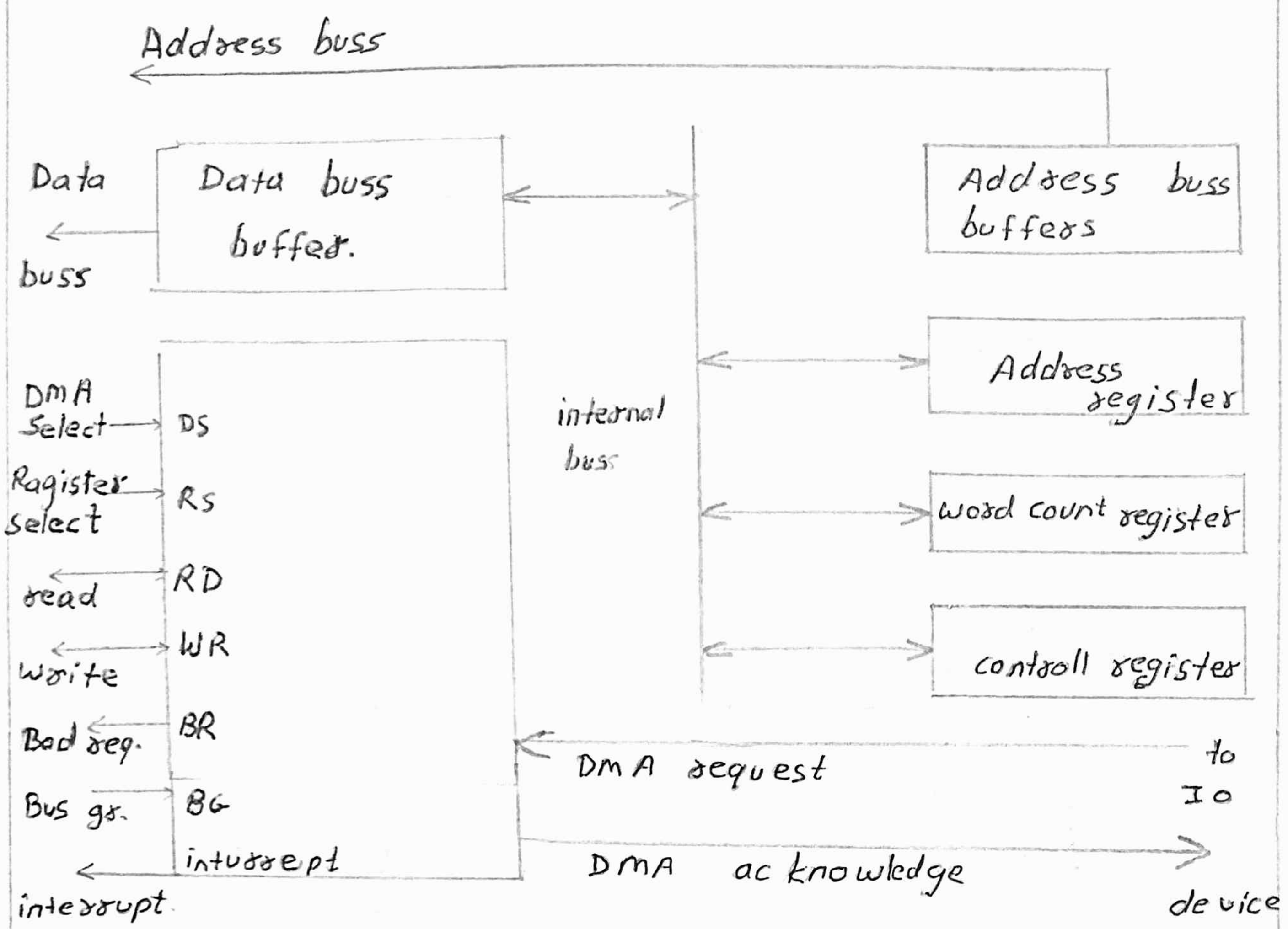
DMA contoller needs some old circuits of an interface to communicate with the CPU and other I/O devices.

Working:

- The unit communicates with CPU through data buss and controll lines
- Through the use of address buss and enabling DMA and RS ragisters selecting inputs, A ragister within DMA is chosen by CPU.
- RD and WR are 2-way inputs
- when BG input is 0, CPU can communicate with ragisters, when BG input = 1, CPU has relenquished the bus and DMA can communicate directly with memory.

Note: All registers in the DMA appear to CPU as I/O interface ragisters, So, CPU can both read and wright DMA ragisters.

20118100



Explanation:

The cpu initialises DMA by sending given info. through data bus.

The starting adress of memory where read/ wright data is available, word count, mode of transfer defined by controll, and controll to begin DMA controll are also sent.

~~Q4~~ Q2 I) immediate addressing:

→ adress field contains the operand itself.

∴ effective adress = ~~301~~ 601

II) relative adressing:

→ address field contains offset to be added to

3/

program counter to adress memory location of operand.

∴ effective adress : 601 + 800 = 1401

III) Register indirect adressing :

→ adress of operand is in the ragister.

adress field contains another operand.

∴ effective adress = R_1 = 400

IV] Indexed adressing with R_1 as index ragister :

→ index absolute adressing. (as there is adress field)

→ effective adress = contents of index ragister + contents of adress field.

= 800 + R1 = 800 + 400

= 1200

Q4 Booth's Algorithm for multiplication :

1. Start
2. Product : 0
3. take input of 2 decimal numbers : n_1, n_2
4. Convert into binary and store as arrays as num_1 and num_2
5. 2's component num_1 & num2 if they are (-)ve
6. 2's component num_2 & Store as ncopy
7. copy num_1 as ncopy

20118100

$41

8. set q = 0

9. if $num_1 = q$: arithmatic shift product ncopy

10. else if $num_1 = 1$ & $q = 0$
→ add ncom to product and arithmatic shift product : ncopy.

11. else add num_2 to product & arithmatic shift product : ncopy

12. in each step, set $q = num_1[i]$ after shift operation

13. Repeat steps 9 - 12 till all bits of num_1 are shifted out.

14. ~~retur~~ return final result as ncopy

Given numbers: multiplicand = 11111 = $(31)_{10}$
multiplier = 10101

signs not included.

Multiplicand in B = 11111

	E	A	Q	SC
Multiplier in Q --	0	00000	10101	101
$Q_n = 1$ add B ---	0	11111 111111		
Shr EAQ ----		01111	11010	100
$Q_n = 0$ Shr EAQ --		00111	11101	011
$Q_n = 1$ add B --	1	11111 00110		
Shr EAQ ----	0	10011	01110	010
$Q_n = 0$, shr EAQ --		01001	10111	001
$Q_r = 1$ add B --	1	11111 01000		

$Q = (21)_{10}$

20118100
51

shr EAQ - - - - $\underbrace{1010001011}_{(651)_{10}}$ 000

Q5 Adress space :

n bits in adress space = $\frac{2^n}{\text{size}}$

adress space bits : 48

Memory space

n bits in memory space ⇒

words in adress space ~~#~~ = 2^n / size of the word

$n = 32$

∴ no. of words in memory space = ~~$2^{16}/16$~~

= 2^{24} =

= $2^{32} / 34k$ =

= 2^{32-5} = 2^{27} = 134217728 words

Q5 Virtual Memory :

The memory that appears to be in main memory (RAM) But, most of it is located in Secondary Storage (like solid state disk) and data is transfered among the two locations as required.

no. of words in adress space = $\underline{2^{48} \text{ bits}}$

no. of words in memory space = $\underline{2^{32} \text{ bits}}$

no. of pages = words in adress space / size of page

= $2^{48} / 4k$ = $\underline{2^{36}}$

2016100 no. of blocks = $2^{32} / 4k$ = $\underline{2^{20}}$

6/

Q7 Given:

time for task by non-pipeliner system = 100 ns

time by pipelined system per task = 20 ns

Stages = k = 6 segment

tasks = 1000 = n

speedup ratio = $S = T_{non-pipelined} / T_{pipelined}$

$= T_{np} / T_p$

also, $T_{np} = 100 \times 1000 = 10^5$ ns

$T_p = (k + n - 1) \times$ time taken by pipelined system

$= (6 + 1000 - 1)\,20 = 20100$ ns

$\therefore$ $S = 10^5 / 2.01 \times 10^4 = \sim 4.975$

i.e. speedup ratio is 4-975

Q6 Given: H (Hit ratio) = 0.8

mem. req. R = 0.9

Cache access time T_c = 50 ns

main mem. access. ~~tm.~~ time T_m = 500 ns

i] Avg. Read access time $= H \cdot T_c + (1 - H)(T_c + T_m)$

$= 0.8 \times 50 + 0.2 \times 550$

$= 40 + 110 = \underline{\underline{150 \text{ ns}}}$

20118100

9i] avr. access time for read and wright

both : $(R \times \text{read access time}) + (1 - R) \times (T_m)$ ns

$= 0.9 \times 150 +$ ~~0.9~~ 0.1×500

$=$ ~~5~~ $135 + 50 = 185$ ns

So, read access time (avr.) is 150 nano-seconds and for read & wright it is 185 nano-seconds.

Q8 Given : $n = 25\,000$ instructions / tasks

$f = 50$ MHz

$k = 5$ stages

processor : 1 instruction issued

i] speedup $S_k = \frac{T_1}{T_k} = \frac{nk\tau}{k\tau + (n-1)\tau} = \frac{nk}{k+n-1}$

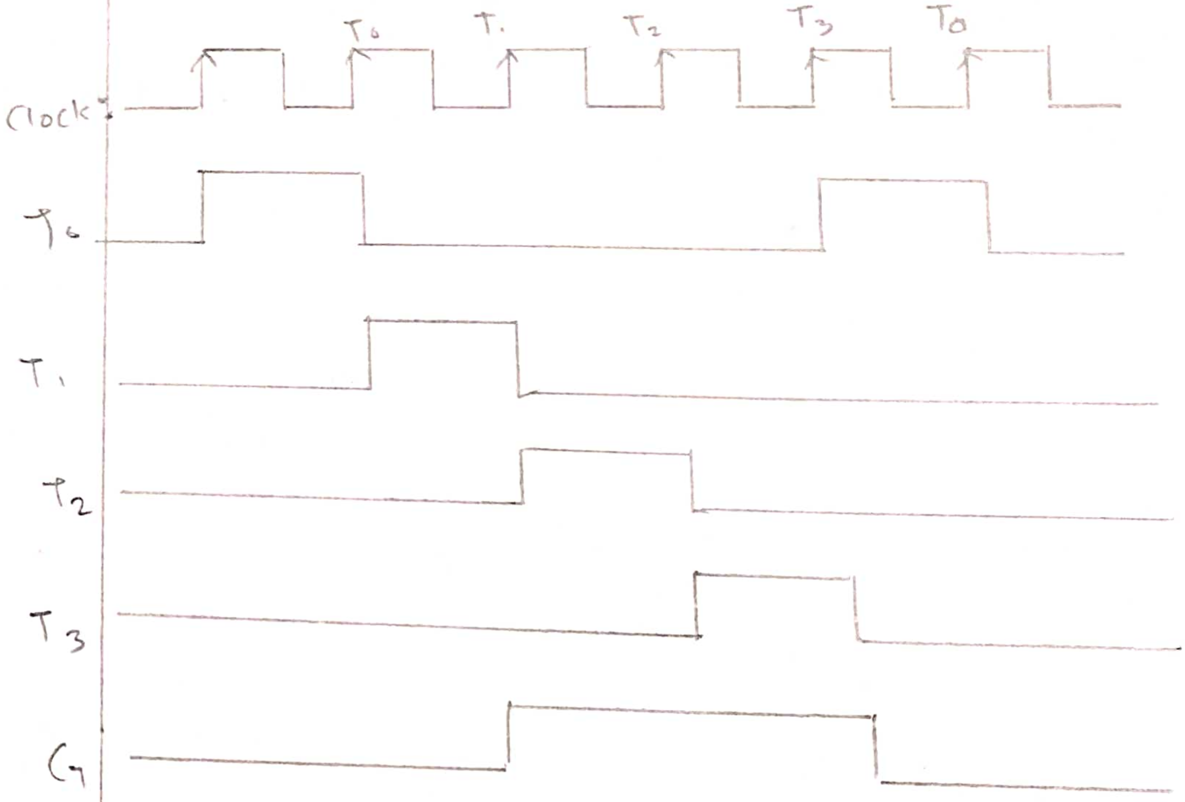
$\therefore S_k = \frac{25000 \times 5}{5 + 25000 - 1} = \frac{1.25 \times 10^5}{2.5004 \times 10^4} \simeq 4.99$

ii] efficiency $= E_k = \frac{S_k}{k} = \frac{4.999}{5} \cong 0.999$

iii] Throughput $= H_k = \frac{nf}{k+n-1} = \frac{25\,000 \times 50 \times 10^6}{5 + 25\,000 - 1}$

$H_k \cong 49.99$ MIPS

Hence, for the execution for the program with 25k instructions, & 50MHz clock rate & 1 instruction / clock and 5 stages, Q10


T0
T1
T2
T3
T0
Clock
T0
T1
T2
T3
C7