1/7 Review Questions from Chapter 13

13-10. How many characters per second can be transmitted over a 57,600 baud line in each of the following modes?

(a) Asynchronous serial transm. with two stop bits:

$$57600 \,/\, (\underset{\substack{\uparrow\\ \text{ASCII}}}{8} + \underset{\substack{\uparrow\\ \text{stop}\\ \text{bits}}}{2} + \underset{\substack{\nwarrow\\ \text{start}\\ \text{bit}}}{1}) = 5236 \text{ chars/sec}$$

(b) For one stop bit:

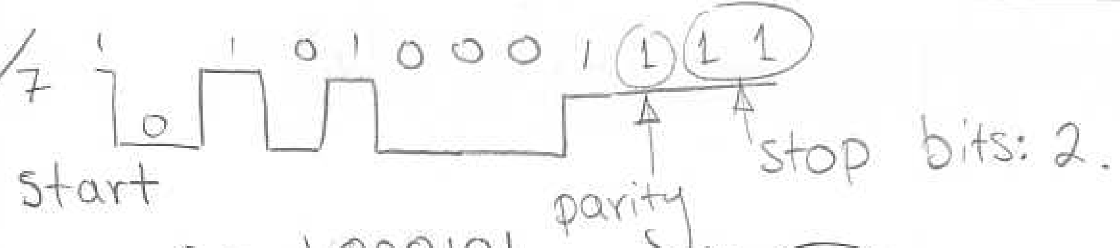
$$57600 / (8+2) \ldots$$

(c) Repeat for 115,200 baud line.

Easy.

13-11. Sketch the timing diagram of the 11-bits that are transmitted over an async. serial communication line when transmitting E. (LSB first + parity)

2/7



$E = \underbrace{1000101}_{3\ 1s}$ ⇒ parity = 1

Note LSB first: 1000101 ⇒ 1010001

13-12. What is the difference between the sync & async serial transfer of info?

Sync. data transm. require:

* receiver + transmitter to sync clocks using SYNC bytes.

* also uses special commands.

Async. data transm. require:

* start + stop bits for comm.

* parity bit for data check.

13-13. Sketch the waveforms for the SYNC pattern used for USB and the corresponding NRZI waveform. Explain why the pattern selected is a good choice for achieving synchronization.

3/7 SYNC = 0000 0001

There are: 7 0's & 1 "1."

Starting from (0) 1010 1011.

Starting from (1) 0101 0100.

} Here, assume that we are starting from 1.

⇒ Many edges to synchronize on.

(13-14) The following stream of data is to be transmitted by USB:

0111 1111 0010 0000

1111 1101 1111 1101

(a) Assuming that bit stuffing is not used, sketch the NRZI waveform.

Ans: Start with 1:

(1) 0000 0000 1001 0101

1111 1100 0000 0010

(b) Modify the stream by applying bit stuffing.

Ans: Start with 1. Apply 0 "before" 7 "1"s.

0111 111$\boxed{0}$ 1 0010 0000

1111 1101 1111 1$\boxed{0}$1 01

4/7 13-15 The 8-bit word "Bye" is to be transmitted to a device address 39 and endpoint 2. List the Output and Data0 packets and the Handshake packet for a Stall for this transmission.

$$39 = 32 + 7 = \overbrace{0100111}^{7\text{-bits}}$$

$$2 = \underbrace{0010}_{4\text{-bits.}}$$

Output packet:

SYNC	Type	check	Device Addr	End	CRC	EoP
8-bits	1001	0110	0100111	0010		

Data packet:

SYNC	Type	Check	ASCII-LSB("Bye")	CRC	EOP
8-bits	1100	0011			

Handshake packet:

SYNC	Type	Check	EoP
	0111	1000	

5/7 13-18 What happens in the daisy chain priority interrupt shown in Fig. 13-15 when device 0 requests an interrupt after device 2 has sent an interrupt request to the CPU, but before the CPU respondr with ACK?

Ans:

	Dev 0				Dev 1				Dev 2			
Description	PI	PO	RF	VAD	PI	PO	RF	VAD	PI	PO	RF	VAD
Initially	0	0	0	—	0	0	0	—	0	0	1	-
Before CPU acks	0	0	1	—	0	0	0	-	0	0	1	-
After CPU acks	1	0	1	0	0	0	0	-	0	0	1	—

13-25 It is necessary to transfer 1024 words from a magnetic disk to a section of memory starting from address 2048. The transfer is by means of DMA as shown in Fig. 13-20.

(a) Give the initial values that the CPU must transfer to the DMA controller.

6/7 Ans:

ⓐ CPU initiates DMA by:
- transferring 1024 to word counter
- transferring 2048 to address reg.

(b) Give the step by step account of the actions taken during the input of the first two words.

① I/O Device sends the DMA controller a "DMA request".

② DMA sends BR to CPU.

③ CPU responds with BG.

④ contents of DMA address reg. on address bus:

- DMA sends "DMA ack" to I/o device

First
- Address 2048+(1024-WCR) on address bus.
- DMA enables the Write control to memory
- Data word is placed on the data bus by I/O device
- Decrement WCR

7/7

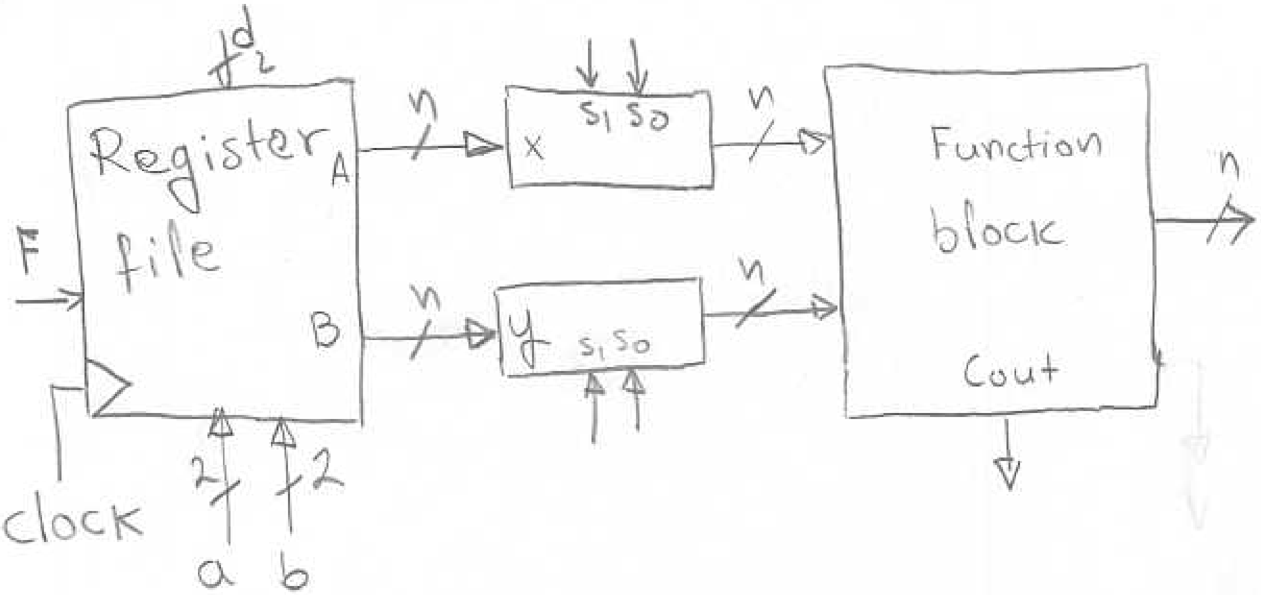
(5) if DMA receives a "DMA req" then repeats (4), else disable BR

(6) CPU reads WCR to determine how many bytes are left to transfer.

Pipeline Problems

This problem refers to a modification of problem 2 in the midterm.

(1) (a) For the architecture blocks below:

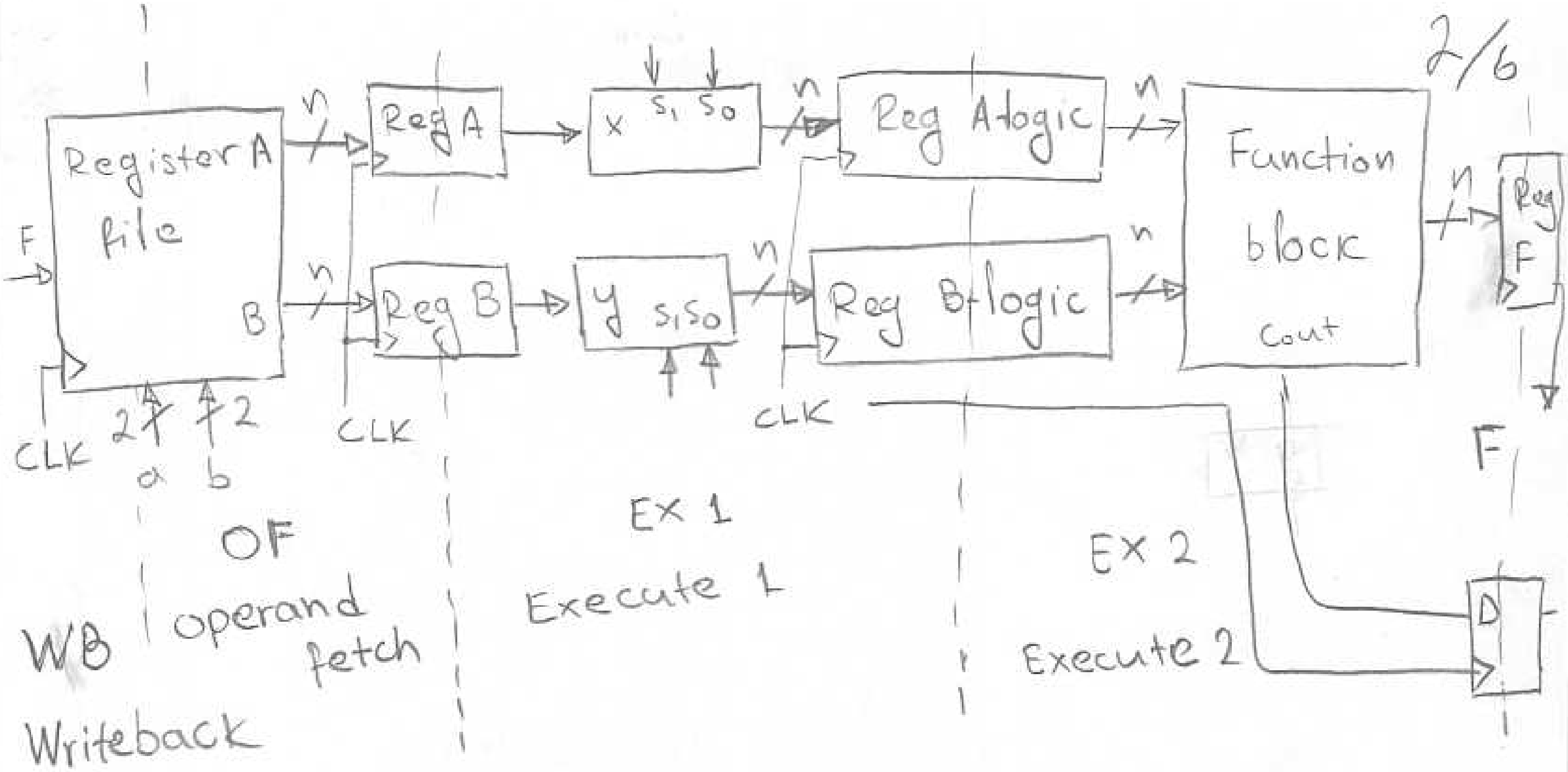


Indicate how we can:

(i) provide a pipelined solution to the problem of speeding up execution.

(ii) for each pipeline stage, please indicate an appropriate name

Ans: The basic idea is to break up the long combinational delay chain by introducing registers and flip-flops.



1 (b) Assume:

* $T_{hold} = 1ns$
* $T_{setup} = 1ns$
* $T_{x\text{-block}} = 2ns$, $T_{y\text{-block}} = 5ns$
* $T_{function\text{-block}} = 8\ ns$, $T_{reg\text{-file-left}} = T_{reg\text{-file-right}} = 2ns$

What are: (i) maximum operating frequency of the original circuit, (ii) maximum operating frequency of the pipelined design.

<u>Ans:</u> (i) Compute the maximum delay:

- We must follow through <u>every</u> possible path:

<u>Path A:</u> Start with the register file.

$$T_A = T_{\text{reg-file-right}} + T_{\text{x-box}} + T_{\text{function-block}}$$

$$+ (T_{\text{setup}} + T_{\text{hold}}) + T_{\text{reg-file-left}}$$

$$= (2 + 2 + 8 + (1+1) + 2)\ \text{ns}$$

$$= 16\text{ns}$$

$$\Rightarrow f_A = 62.25\ \text{MHz}$$

<u>Path B:</u> $T_B = T_A - T_{\text{x-box}} + T_{\text{y-box}}$

$$= 16\text{ns} - 2\text{ns} + 5\text{ns}$$

$$= 19\text{ns}.$$

$$\Rightarrow f_B = 52.632\ \text{MHz}$$

$$\Rightarrow f_{\text{original}} = \min(f_A, f_B)$$

$$= \underline{\underline{52.632\ \text{MHz}}}$$

(ii) There are several stages:

* WB: $T_{WB} = T_{reg\text{-}file\text{-}left} + (T_{hold} + T_{setup})$

$= 2ns + 1ns + 1ns = 4ns$

* OF: $T_{OF} = T_{reg\text{-}file\text{-}right} + (T_{hold} + T_{setup})$

$= 4ns$

* EX1: $T_{EX1,A} = (T_{hold} + T_{setup}) + T_{x\text{-}block}$

$= 2ns + 2ns = 4ns$

$T_{EX1,B} = (T_{hold} + T_{setup}) + T_{y\text{-}block}$

$= 2ns + 5ns = \underline{\underline{7ns}}$ largest delay in EX1.

* EX2: $T_{EX2} = (T_{hold} + T_{setup}) + T_{function\text{-}block}$

$= 2ns + 8ns$

$= \underline{\underline{10ns}}$.

$T_{max} = max(T_{WB}, T_{OF}, T_{EX1,A}, T_{EX1,B}, T_{EX2})$

$= \underline{\underline{10ns}}$

$\Rightarrow f_{pipelined} = \underline{\underline{100 MHZ}}$

1ⓒ Show the execution of:

ADD R3, R1, R2
SUB R0, R1, R2

Ans:

Clock cycle

	1	2	3	4	5	6	7	8	9	10	11
R3 ← R1 + R2	OF	EX1	EX2	WB							
R0 ← R1 − R2		OF	EX1	EX2	WB						

1ⓓ When the pipeline is fully utilized and there are no data dependencies, what is the speedup?

Ans: This is the ideal case where:

* we execute one instruction per cycle

So, speedup $= \frac{19ns}{10ns} = \underline{\underline{1.9}}$

1ⓔ Consider:

ADD R3, R1, R2
SUB R2, R3, R1

How do we fix execution is software?

Clock Cycle

6/6

	1	2	3	4	5	6	7	
R3 ← R1 + R2	OF	EX1	EX2	WB				
NOP		OF	EX1	EX2	WB			
NOP			OF	EX1	EX2	WB		
NOP				OF	EX1	EX2	WB	
R2 ← R3 - R1					OF	EX1	EX2	WB
						OF	EX1	EX2

1(f) Show solution in hardware & software by modifying the pipeline ...