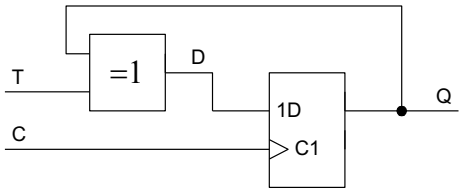
ELEC50001 EE2 Circuits and Systems

Problem Sheet 7 Solutions –Timing Constraints & Memory *(Lecture 14 & 15)*

(Question ratings: A=Easy, …, E=Hard. All students should do questions rated A, B or C as a minimum)

1B. As seen in problem sheet 1, an XOR gate can be used to invert a signal or pass it through unchanged according to whether a control input is high or low.



2C. We define t=0 as the falling edge of CA.

Setup requirement: $\max(DB\uparrow\downarrow)+12 < \min(CB\uparrow)$
$50+22+12 < (13 + \frac{1}{2}T)$
$\frac{1}{2}T > 71 \Rightarrow f < 7$ MHz

Hold requirement: $\max(CB\uparrow) + 27 > \min(T + DB\uparrow\downarrow)$
$\frac{1}{2}T+22 + 27 > T + 5+13$
$\frac{1}{2}T > 31$ (less severe restriction than above)

Note the extra T term in the hold requirement: this is because we want the *second* transition of DB to occur >27 ns after CB↑. The Hold requirement is so easily satisfied that it wouldn't normally be necessary to calculate it exactly.

3C.

µPA → flipflop CA↓=0	Setup:	$\max(DA\uparrow\downarrow) +5 < \min(CA\uparrow)$ $50+5 < \frac{1}{2}T$ $\frac{1}{2}T > 55 \Rightarrow$ **<u>f < 9 MHz</u>**	Hold:	$\max(CA\uparrow)+3< \min(T+DA\uparrow\downarrow)$ $\frac{1}{2}T+3< T + 5$ $\frac{1}{2}T > -2$ ☑
flipflop → flipflop CA↑=0	Setup:	$\max(DB\uparrow\downarrow)+5 < \min(CB\downarrow)$ $10+22+5 < \frac{1}{2}T+13$ $\frac{1}{2}T > 24 \Rightarrow f < 21$ MHz	Hold:	$\max(CB\downarrow)+3< \min(T+DB\uparrow\downarrow)$ $\frac{1}{2}T+22+3< T +2+13$ $\frac{1}{2}T > 10 \Rightarrow f < 50$ MHz
flipflop → µPB CB↓=0	Setup:	$\max(FB\uparrow\downarrow)+12<\min(CB\uparrow)$ $10+12<\frac{1}{2}T$ $\frac{1}{2}T > 22 \Rightarrow f < 23$ MHz	Hold:	$\max(CB\uparrow)+27<\min(T+FB\uparrow\downarrow)$ $\frac{1}{2}T+27<T + 2$ $\frac{1}{2}T > 25 \Rightarrow f < 20$ MHz

It can be seen that the critical figure is the setup time for the first flipflop: this is because the microprocessor takes such a long time (up to 50 ns) to output its data.

Question 4 (which doesn't work) and question 5 (which does) show how to relax this constraint. In the third row of the previous table, I have cancelled out the delay of the clock line driver/receiver from the two sides of the inequality. This is only valid if we can assume that the propagation delays for rising and falling edges are the same (not generally true).

4C. The first flipflop now responds to a falling clock edge: this means that µPA now has a full clock cycle to output its data rather than only a half cycle. We have therefore doubled maximum clock frequency of the circuit. (Note that the middle row of this table is unchanged from the previous question).

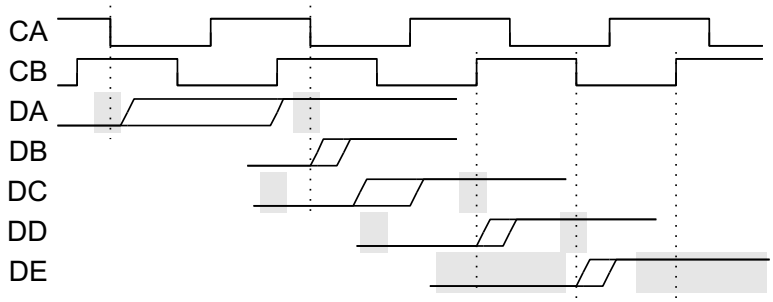
The problem is that the output from the second flipflop now changes on the rising clock edge and therefore fails to meet the hold time of µPB.

µPA → flipflop CA↓=0	Setup:	max(DA↑↓)+5<min(T+CA↓) 50+5<T T > 55 ⇒ **f < 18 MHz**	Hold:	max(CA↓)+3<min(DA↑↓) 0+3<5 ☑
flipflop → flipflop CA↓=0	Setup:	max(DC↑↓)+5<min(CB↑) 10+22+5 < ½T+13 ½T > 24 ⇒ f < 21 MHz	Hold:	max(CB↑)+3<min(T+DC↑↓) ½T+22+3<T + 2+13 ½T > 10 ⇒ f < 50 MHz
flipflop → µPB CB↑=0	Setup:	max(DD↑↓)+12<min(T+CB↑) 10+12<T T > 22 ⇒ f < 46 MHz	Hold:	max(CB↑)+27<min(DD↑↓) **2 > 27** ☒

5D. We can fix the hold problem by adding a third flipflop. The last row of the previous table is now replaced by the two rows below and the maximum frequency is now 18 MHz.

flipflop → flipflop CB↑=0	Setup:	max(DD↑↓)+5<min(CB↓) 10+5<½T ½T > 15 ⇒ f < 33 MHz	Hold:	max(CB↓)+3<min(T+DD↑↓) ½T+3<T + 2 ½T > 1 ⇒ f < 500 MHz
flipflop → µPB CB↓=0	Setup:	max(DE↑↓)+12<min(CB↑) 10+12<½T ½T > 22 ⇒ f < 23 MHz	Hold:	max(CB↑)+27<min(T+DE↑↓) ½T+27<T + 2 ½T > 25 ⇒ f < 20 MHz

The timing of this circuit with a clock period of about 60 ns (16.7 MHz) is shown below with setup/hold windows shaded:



6A. Two reasons. Firstly many memories, though not all, use the same pins for data input as for data output: the outputs must therefore be turned off (or "tristated") to allow new data to be written into a memory location. Secondly, a large memory system contains several memory integrated circuits which are enabled one at a time according to the address range selected. The use of tri-state outputs allows all memory data lines to be connected together without the need for an external multiplexer to switch between them.

7B. a) RAM has 13 address inputs, ROM has 14, Serial Port has 3 and Parallel Port has 1.

b) We need 5 RAM chips and one ROM, serial and parallel chips. The CE inputs are given in the following table:

Chip	Address Range	CE
RAM	0000 – 1FFF	$\overline{A15}\cdot\overline{A14}\cdot\overline{A13}$
	2000 – 3FFF	$\overline{A15}\cdot\overline{A14}\cdot A13$
	4000 – 5FFF	$\overline{A15}\cdot A14\cdot\overline{A13}$
	6000 – 7FFF	$\overline{A15}\cdot A14\cdot A13$
	8000 – 9FFF	$A15\cdot\overline{A14}\cdot\overline{A13}$
ROM	A000 – DFFF	$A15\cdot\left(\overline{A14}\cdot A13 + A14\cdot\overline{A13}\right)$
Serial	E100 – E107	$A15\cdot A14\cdot A13\cdot A8$
Parallel	E200 – E201	$A15\cdot A14\cdot A13\cdot A9$

Note that I have not included all the address lines needed to fully decode the Serial and Parallel port adress ranges. The microprocessor should never access the undefined memory locations in the range E000 to FFFF so it does not matter if the peripheral ports respond to several of them. As defined above, the following addresses will all refer to the serial port's lowest location: E100, E108, E110, E118, …, FFF8. Of the 16 address lines, 3 are direct inputs to the serial port, 4 are used in forming its CE and 9 are unused. The 9 unused lines can take on 512 possible values and so the serial port will appear 512 times in the memory map. If a PAL is being used to generate the CE signals, then the number of PAL inputs required may be reduced by not decoding peripheral address ranges fully.

c) The bytes are in the wrong order in the ROM. The ROM has 14 address inputs, namely A13:0. Addresses A000 to BFFF have A13=1 and will therefore be mapped to the second half of the ROM; addresses C000 to DFFF have A13=0 and will be mapped to the first half of the ROM. This situation could be corrected by inverting A13 before connecting it to the ROM but this would add delay: a neater solution is to use A14 as the most significant address bit rather than A13.

8B. Note that only the setup time matters for this question. Let A be the access time (or propagation delay) from the address inputs and E be the access time from the OE input. The clock period is 50 ns. P is the propagation delay of a gate (i.e. 1 to 2 ns)

$$\max(7+A)+3 < 50 \Rightarrow \underline{\mathbf{A < 40\ ns}}$$
$$\max(P+E)+3 < 25 \Rightarrow \underline{\mathbf{E < 20\ ns}} \text{ (taking P=2, its maximum value)}$$

9A. If the same wire is driven high by the output of one device and low by that of another, a high current will flow which will waste power and which may even damage the integrated circuits involved. To reduce the possibility of two devices trying to drive the same wire simultaneously, tristate outputs are almost always designed to turn off more quickly than they turn on.