

Digital Design: exercise session 4

WeiJie Jiang, Ce Ma, Wim Dehaene

January 16, 2025

1 Introduction

For now, we have focussed on the logic and logic optimisations in the microprocessor, originally introduced in Session 1 (see figure 1). However, a microcontroller also requires memory elements: (1) flipflops, to create a synchronous digital chip and (2) a memory.

This session will first analyse the timing of a TSPC register in a more theoretical way. Then, with this timing information, we can calculate the clock speed of the microprocessor. Afterwards, the basics of SRAM design are touched. Finally, testability is considered, which is of utter importance for large volume production.

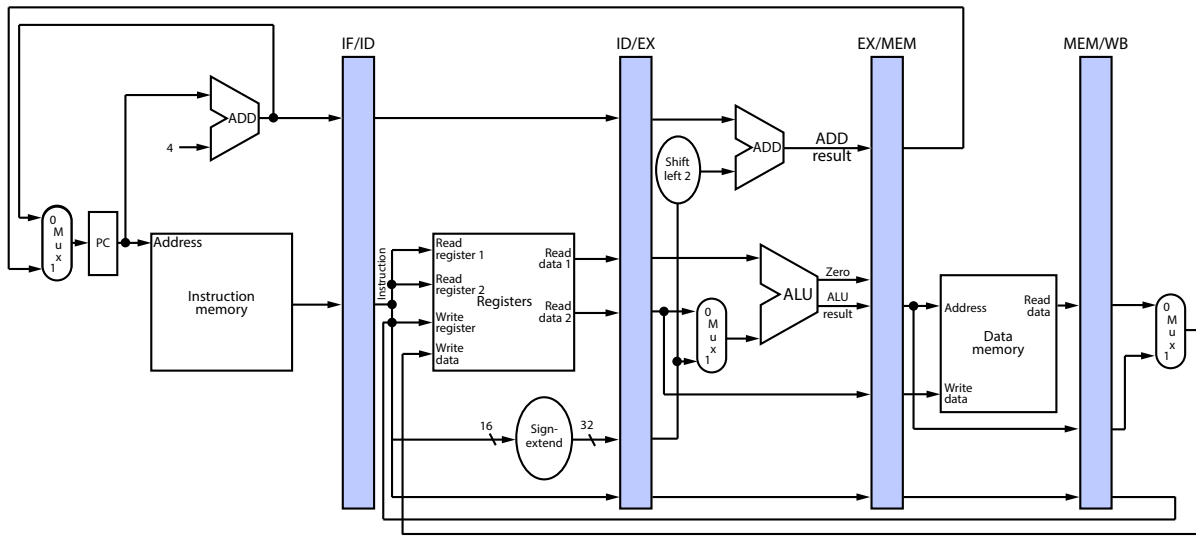


Figure 1: 5-stage pipelined microprocessor architecture

2 Register

The TSPC master slave register shown in Figure 2. This section only uses hand calculations.

1. Explain how it works.
2. Scale all transistors so that every stage has the same drive strength as a minimal inverter.
3. Calculate the relative setup time (t_s) and propagation delay (t_{cq}). Assume the output of the register is loaded with an inverter of 4 times minimal size.

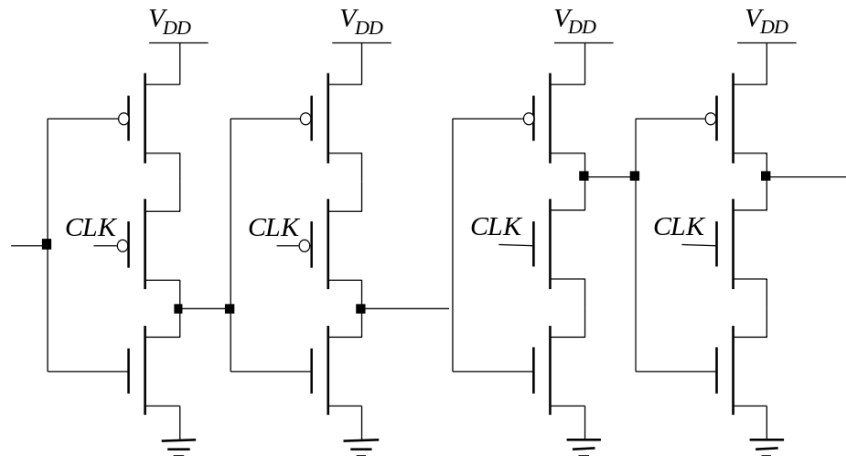


Figure 2: TSPC register topology

3 Microprocessor clock speed

As you all know, the clock speed is determined by the critical path delay in the logic section between two flipflops. However, some extra delays need to be taken into account:

- When the clock arrives at the register, it takes some time for the new value to propagate to the output (t_{cq})
- A signal needs to be stable for some time before the clock launch, as it will not be propagated any more (setup time, t_s).
- In a physical design, the clock edge does not launch exactly at the same time on all flipflops, as there is some skew on the clock tree. Take 5% margin for clock tree variation. Additionally, clock tree skew can induce hold violations, which we will not consider here.

Remember from previous exercise sessions:

- The intrinsic delay of the inverter (Session 1) is $t_{p0} = 11.9\text{ps}$.
- The critical path delay of the brent-kung adder (Session 3) is $t_{logic} = 952.4\text{ps}$.

Now, answer the following questions:

1. Calculate absolute setup time and propagation delay, using the intrinsic delay (t_{p0}) from the inverter.
2. Calculate the total path delay considering all delay sources.
3. What is the max. operating speed, including the 5% margin for clock skew?

4 SRAM cells

This section focuses on the memory of the processor. Here, we will design a single cell of the memory to get a basic idea of the characteristics of a memory (area, leakage power, ...). Consider the 6T SRAM cell from Figure 3. It consists of two crossly coupled inverters and access transistors to read and write the data in the internal nodes Q and \overline{Q} .

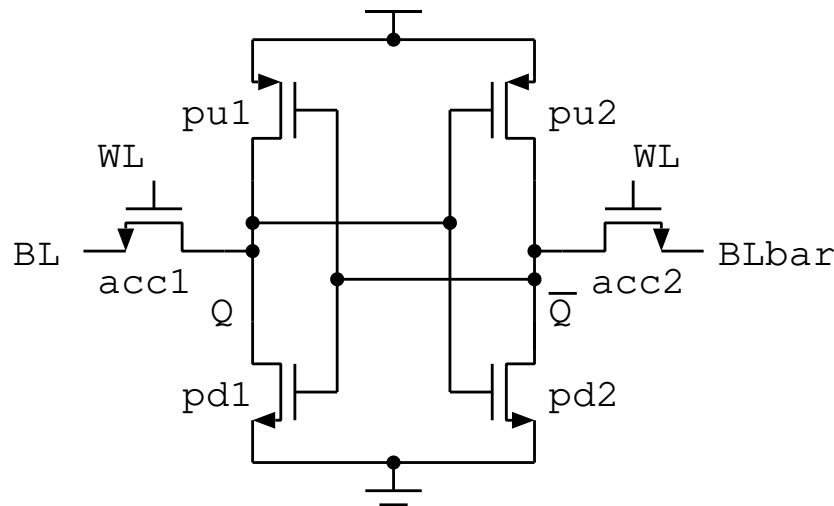


Figure 3: 6T SRAM cell

1. Explain the different behaviour when the word line WL acquires the value 0 and 1.
2. Would you use high V_T or low V_T transistors for SRAM cells? Why?
3. Size all transistors using the parameters for 45nm predictive technology from table 1, the assumptions summarized in table 2 and the following step-by-step guide. Fill in the sizing of the transistors in “butterfly.m2s”.
 - $V_{DD} = 1V$;
 - To enable the access transistors without the need of a pass gate, the wordline voltage is boosted 300mV above the supply.
 - The memory cell should have minimum area and leakage. Thus, L is chosen above the minimum length: $L = 80nm$. Additionally, it is preferred to size as many transistors as possible to minimum size ($W=120nm$). In this case, choose $W_{n,wl} = 120nm$;
 - The cell should be writable: To write a '0', the internal node (V_Q) should be pulled down to ground to minimum $\frac{V_{DD}}{8}$, in order to let the other side flip upwards. Write down the appropriate current equations and find the W_p for the pull-up transistor.
 - The cell should be readable: Read upset should be avoided: the internal node (V_Q) that stores a '0' should be raised maximum to $\frac{V_{DD}}{8}$. Write down the appropriate current equations and find the W_n for the inverter's nmos pull-down transistor.
4. The memory cells typically take around 40% of the total memory area. Due to metal routing, the cell area is around 20 times larger than the active area of the transistors. Using these two numbers, estimate the total area of a 128kb SRAM.

5 Test

Remember the adder that was used in the proposed microprocessor from figure 1. As you can see, the inputs and outputs of the adder are connected to flipflops.

1. What would you do to modify the architecture to become fully testable?
2. Use the D-algorithm to find test vectors to detect Sa0 and Sa1 for node s2. Hint: the involved gates are drawn in figure 4.

parameters	45nm HP		45nm LP		Parameter	Value
	nMOS	pMOS	nMOS	pMOS		
V_t (V)	0.339	-0.335	0.561	-0.515	V_{DD} (V)	1
$V_{D,sat}$ (V)	0.396	0.457	0.290	0.358	V_{WL} (V)	1.3
K' ($\mu A/V^2$)	1500	520	930	400	V_Q	$\frac{V_{DD}}{8}$
λ (1/V)	0.55	0.7	0.55	0.7	Table 2: SRAM as- sumptions	
W_{min} (nm)	120	120	120	120		

Table 1: Transistor parameters in 45 nm

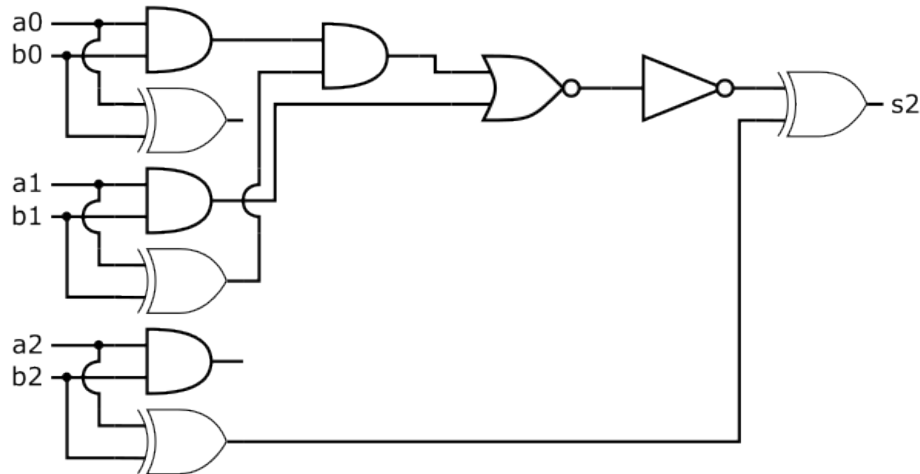


Figure 4: reduced circuit to test sa faults for s2

6 Extra SRAM questions

If you have extra time, we can simulate the SRAM cell, based on your findings of section 4. With these measurements, stability and power estimations can be made in a much more reliable way.

The setup is similar to previous sessions:

```
cd ~/Documents/DigitalDesign/
cp -rP ~cnieta/Public/DDIC/DD_Es4 DD_Es4
cd DD_Es4
```

Start matlab in this folder.

1. **Stability measurement.** A key figure of merit is the Static Noise Margin (SNM), meaning the quantity of DC noise required to flip a cell. It can be measured using the butterfly curves, where the SNM is defined as the side-length of the largest embedded square on each 'eye' (Seevinck square). Evaluate the stability of your design using the butterfly curves generated by *butterfly.m*. Does this leave enough room for variation?
2. Check the read stability by enabling the *globals.wl_voltage* to the correct voltage in *butterfly.m*. Why does the graph change this way?
3. The *butterfly.m* script also shows the leakage power of one cell. What is the leakage power of 128×1024 cells?