

ELE447 Lab 5: Latches, Flip-Flops & Registers (2-week assignment)

At the core of every static memory cell is a cross coupled pair of inverters or simple inverting gates (e.g. NAND and NOR gates). Figure 1 depicts the simplest version of a 1 bit static memory consisting of 2 inverters.

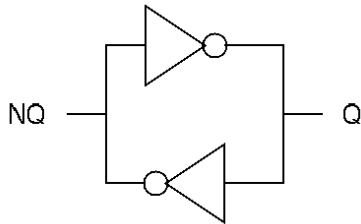


Fig1. 1-bit static memory cell.

If the 2 inverters are complemented by 2 transmission gates, which provide access to the complementary outputs, one obtains the basic 6 transistor static Random Access Memory (**sRAM**) cell (to save area, the two pass transistors in the sRAM are realized by two single n-channel devices).

If the cell in figure 1 is complemented by a single pass transistor to enable writing, it is called a **Latch**. Note that each time the latch output needs to be changed, a conflict arises between the inverter in the feedback path and the input driver. The feedback element tries to preserve the output while the input driver wants to change the state. To prevent this *tug-of-war* scenario, the feedback loop is often cut open during the write phased. Such a *conflict free* latch circuit is shown in figure 2.

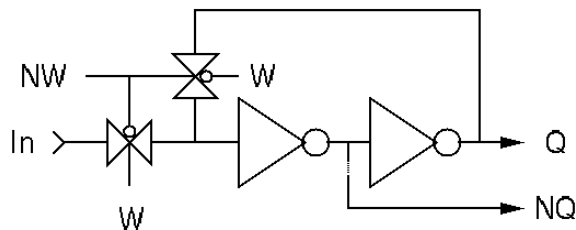


Fig2. CMOS latch with open feedback loop during write mode.

It is important to note that the latch in figure 2 acts similar to a transmission gate if W is affirmed, i.e., $Q=In$. Therefore, it is often referred to as a semi-transparent latch.

To prevent a direct feed-through between input and output, a second latch is required, which operates on the complementary write signal (master/slave configuration). Such a cell still equates the output to the input, but delays the state transition until the falling edge of control signal W . If W is realized by a clock signal, such a composite memory element is called a synchronous D Flip-Flop (FF), where the letter D stands for delay. There are alternative implementations for D-FFs, but the double-latch version employing CMOS transmission gates is very compact and thus frequently used by digital circuit designers.

Tasks:

- Run Magic and load the cell *sdf1_50* (a static D-FF). Create the transistor level schematic of this library cell.
- Use **Hspice** to verify the proper functionality of the cell *sdf1_50*. Make sure you simulate all possible combinations between input signal D and control signal Clk.
- Create a **4-bit shift register** using the cell *sdf1_50* as a building block. Your layout needs to be hierarchical, in other words, use *sdf1_50* as a sub-cell.
- Verify the proper functionality of your 4-bit shift register by means of **Irsim**.
- Create the schematic for a **4-bit shift register**, which can be loaded in parallel and shift the data out serially. To do so, you need an additional control input Ld (Ld=1 load parallel, Ld=0 shift bits from left to right) and some additional *glue* logic.
- Realize your parallel/serial register in Magic. Use existing library cells whenever possible.
- Verify the proper functionality of your serial-parallel register using **Irsim**.
- Run Magic and load the cell *stff2_50*, a static asynchronous toggle (T) FF. Compare this cell with *sdf2_50*. What is the difference between these 2 cells?
- Realize a **4-bit asynchronous counter** using the cell *stff2_50* as a building block.
- Verify the proper functionality of your counter using **Hspice** and **Irsim**. Specifically, observe the exact transition times of your 4 outputs. Why do we refer to this implementation as an asynchronous circuit?
- Modify your 4-bit counter such that it only counts the 10 decimal digits 0 - 9, i.e., it resets itself after count 9. Such a counter is called a *modulo-10* counter. (Hint: begin the design procedure by defining the reset condition and subsequently realize the logic that implements the reset). Note: Your reset logic is likely to require another memory element!
- Verify your *modulo-10* counter using **Irsim**.