VE270 Introduction to Logic Design

Recitation Class 9

CHEN Yuchi

citrate@sjtu.edu.cn

July 29, 2020

Part 1. Arithmetic Components

Incrementor

Two ways to design.

1. By truth table: Small, fast, hard to design
2. By adder: Large, slow, easy to design

If the bit width is small, former type may be used. For a large number, latter method will be much faster.

Comparator

Equality comparator: XNOR gates and AND gate. Of course you can use truth table but using XOR gates and AND gate is easier to design and has a fixed delay.

Magnitude comparator: "carry-ripple style" Of course there's no carry. The conditions for output can be easily obtained by common sense.

```
out_gt = in_gt + (in_eq * a * b')
out_lt = in_lt + (in_eq * a' * b)
out_eq = in_eq * (a XNOR b)
```

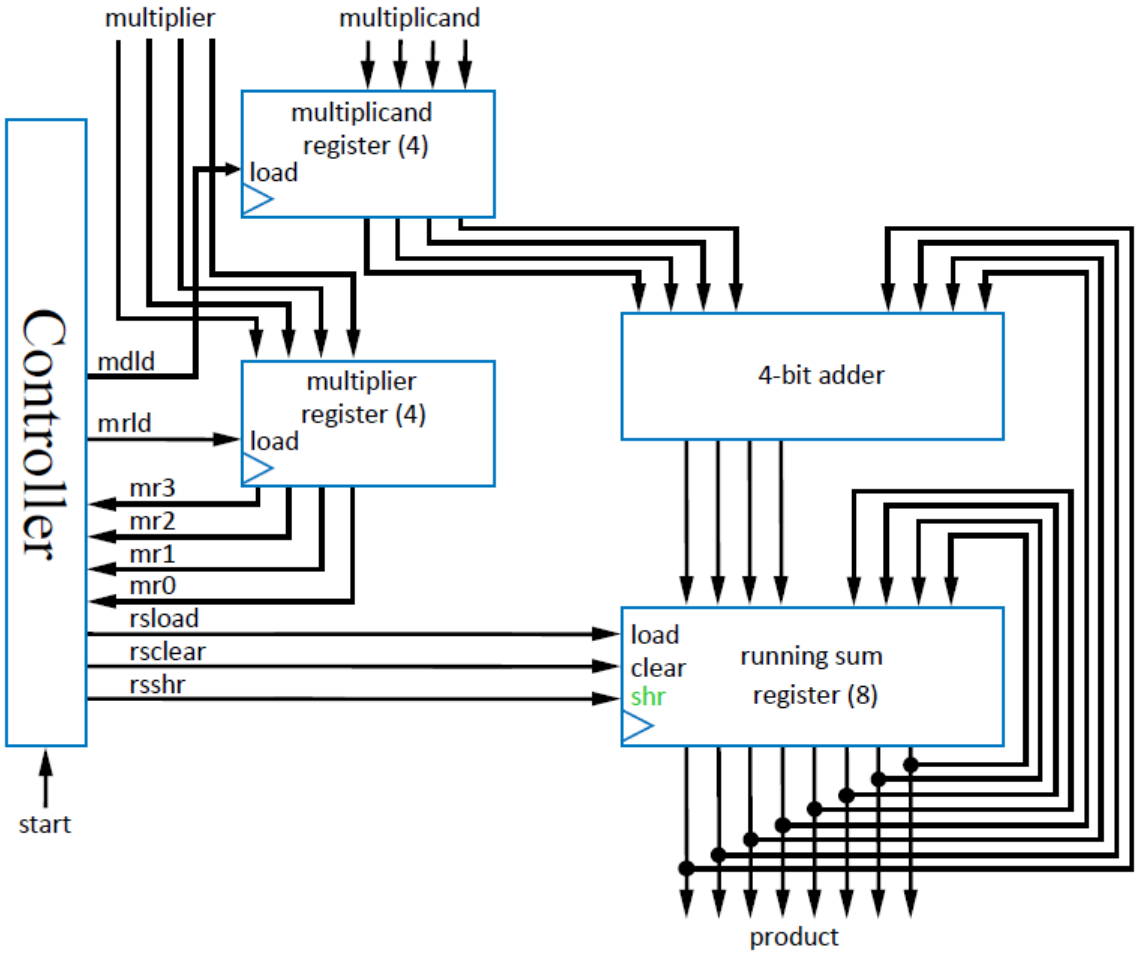
gt means greater than, lt means less than, eq means equal to.

Multiplier

You have learned that if you want to multiply a number with a fixed number, you can use shifter and adder. Here, we discuss about multiplying two random number.

The fundamental rule for binary multiplication is that a number multiplies with 1 is just repeating itself. Then, the first way to design a multiplier is to use basic gates and adder. The delay is fixed and since it's a combinational circuit, it works fast. However, with the bit-width increasing, it can become very large.

Another way is to use a HLSM. The shift register is the core component.



Part 2. Timing Issue

Setup Time and Hold Time (ns)

You can understand it as the data input to a register must arrives earlier than the rising edge by some time and change later than the rising edge by some time, or the register will enter metastable state.

A common signal violates such rules are the input signal (switch, button, keypad and so on). An external input is **unpredictable and may have glitches and fanout pathway delay,** Usually a circuit works highly depending on the clock, so the input needs to be synchronized.

Even cascade synchronizers cannot ensure a 100% synchronized signal. Usually one or two flip-flop(s) are enough.

Switch/Button Debouncing (ms)

SR Latch: Assume the switch will never be released.

Flip-flop: Switch or button may be released. The clock period should be carefully design. If you use flip-flop, you can synchronize the input at the same time.

Clock Skew

Maybe caused by gated clock and the difference in length of clock path.

Usually we don't use gated clock. The reason is to avoid clock skew.

Be can use H-tree to ensure clock signal reaches different part of the circuit simuteanously.

If the paths cannot have an equal length, we have to add buffer to the shorter paths.

Hazards

There are **static 1-hazard** and **static 0-hazard**.

Static hazards are caused by moving from one PI to another. Static 1-hazard can be fixed by including an redundant PI. Static 0-hazard cannot be fixed sometimes.

You can find how to fix static 0-hazard in the other file.

Dynamic hazard is caused by multiple reconvergent paths. It can be fixed by eliminating all of the static hazards.

Functional hazard happens when two or more inputs change together, It cannot be fixed logically.

Part 3. RAM

Terms

Byte: 8 bits.

HW or Half Word: 16 bits.

Word: 32 bits.

An $M \times N$ RAM: M words (rows), N bits width. (Thus, usually $N = 32$)

SRAM

The logic of RAM is similar to register file. It use an address to locate the cell that you want to read or write. There's also an rw signal telling the cell you want to read it or write it.

In fact, a cell needs some additional component (besides transistors) to be read. We only focus on the transistor part.

The core structure is two NOT gates and two nMOS. The two NOT gates can "lock" the value.

When you want to read it, give both sides a high voltage and monitor the change of voltage.

DRAM

Much simpler structure: One transistor, one capacitor.

The capacitance needs to be large enough to store the voltage for enough time. Thus, capacitors are usually on another chip or die.

Capacitor loses charge over time. Thus, it needs to be re-written. That's why it's called "dynamic" RAM.

Comparison

Register File	SRAM	DRAM
Fastest	Fast	Slowest
Small capacity large size	Compact	Very compact