EECS151 : Introduction to Digital Design and ICs

Lecture 20 – Dividers

Bora Nikolić

Pentium FDIV Bug (from Wikipedia)

The Pentium FDIV bug is a hardware bug affecting the floating-point unit (FPU) of the early Intel Pentium processors. Because of the bug, the processor would return incorrect binary floating point results when dividing certain pairs of high-precision numbers. The bug was discovered in 1994 by Thomas R. Nicely, a professor of mathematics at Lynchburg College. Missing values in a lookup table used by the FPU's floating-point division algorithm led to calculations acquiring small errors. While these errors would in most use-cases only occur rarely and result in small deviations from the correct output values, in certain circumstances the errors can occur frequently and lead to more significant deviations

Review

- Binary multipliers have three blocks:
  - Partial-product generation (NAND or Booth)
  - Partial-product compression (ripple-carry array, CSA or Wallace)
  - Final adder
- Multipliers are often pipelined
- Constant multipliers can be optimized for size/speed
- Shifters and crossbars are common building blocks in digital systems
  - Often require customization

Administrivia

- Homework 8 due this week
  - In scope for midterm
- All labs need to be checked off by today!
- Project checkpoint 2 next week
- Midterm 2 is on November 4 at 7pm
  - Review session tomorrow
  - Up to today's lecture
  - 1 page of notes allowed

Dividers

Pencil-And-Paper Division

	1512	quotient
	3 | 4537	dividend
divisor	3000	divisor*q_i*10^i
	1537	partial remainder
	1500	
	0037	
	0030	
	0007	
	0006	
	0001	remainder

Division is an iterative process:

$$\boxed{r(i) = r(i+1) - q_i * D * 10^i}$$

We usually 'guess' q_i

Restoring Divider

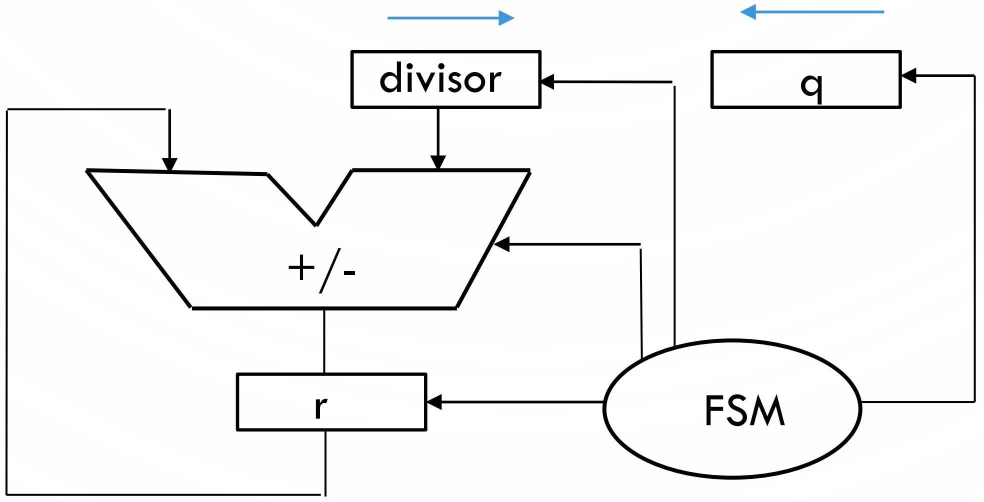
- Assume $q_i = 1$
- Subtract divisor from r; check if r(i) >= 0
  - if r(i) > 0, guess was good ($q_i = 1$)
  - if r(i) < 0, restore the value by adding divisor, $q_i = 0$
- Shift divisor to right
- Repeat *n* times

More efficient to shift the reminder right

n shifts

n subtractions

n/2 restorations



Non-Restoring Divider

- Doesn’t restore if r(i) < 0
- Instead, adds the divisor in the next iteration
  - *n* shifts
  - *n* additions/subtractions

Faster Dividers

- Divide in a higher radix than 2 (typically 4, i.e. guess $q_i q_{i+1}$)
- Keep the partial remainders in redundant form
- Sweeney-Robertson-Tocher (SRT) algorithm
  - Used in many processors

Review

- Binary division is a slow, iterative process
- Non-restoring division speeds it up
- SRT divider, higher radix, redundant number representation