Processors & Pipelining Intro

Scott Beamer (cs61c-ta)

Datapath Components

- In the following problems, draw an implementation of the desired component
- *Extender* for the immediate in MIPS

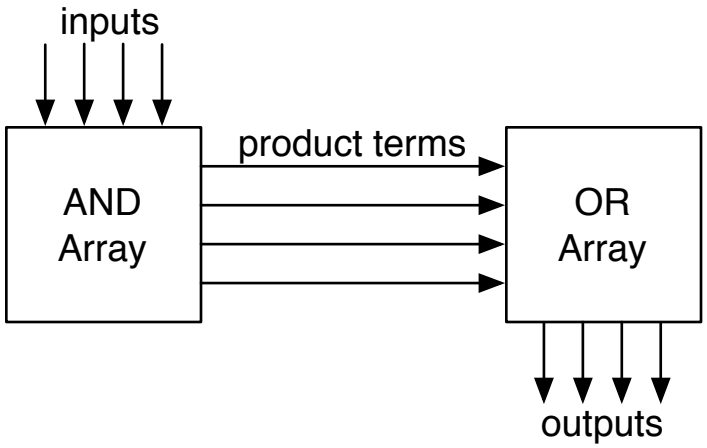
- *Flip-Flop* - with an EN (enable), with a RST (reset), and with both (give priority to reset

- *Register File* - with: 4 registers, 2 read ports, 1 write port, 1 WE (write enable)

- *Instruction Fetch Unit* - Supporting MIPS operations: PC+4, branch, jump, and jr. Assume you have a black box for Instruction Memory.

Programmable Logic

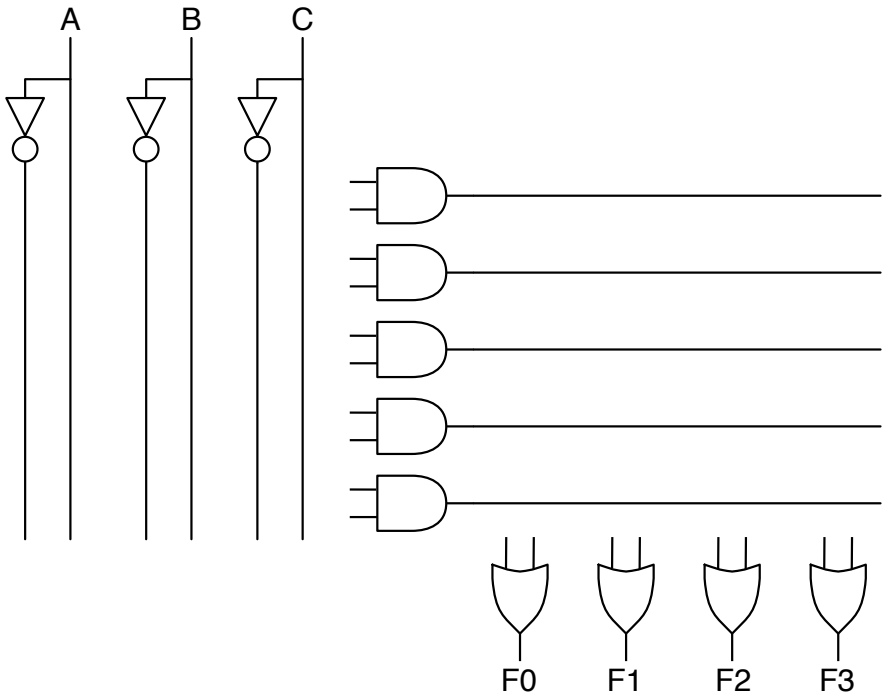
- Programmable logic can be manufactured as standard components, but can be configured to implement arbitrary logic
- This is often useful for control since it is usually specific to the circuit it is designed for
- Programmable Logic Arrays (PLA) - Use sum-of-products since it a convenient form
- Feeds all inputs into an array of AND gates, and then feeds the product terms into an array of OR gates



PLA Example

- Implement F with the PLA

A	B	C	F
0	0	0	0101
0	0	1	1000
0	1	0	0000
0	1	1	0000
1	0	0	1111
1	0	1	1001
1	1	0	1111
1	1	1	1111



Pipelining Introduction

- It improves the *throughput*, not the *latency* of an individual instruction. This means that a single instruction will take the same amount of time or even longer, but over a period of time a pipelined processor can get more done
- *Latency*: The delay to complete the operation from start to finish (a time)
- *Throughput*: How much computation can be done over time (a rate)
- The improvements of pipelined processors can be seen as:
  - Reducing the critical path by inserting registers into the circuit that will break a larger operation into stages. A shorter critical path means a higher possible clock frequency.
  - Utilizing more of the hardware at a given time. Without pipelining, much of a logic circuit spends most of its time in steady state.

Pipelining Hazards

- *Structural Hazards:* The same component is needed at the same time in multiple stages
- *Control Hazards:* Which instruction is next isn't known until after it needs to be fetched
- *Data Hazards:* The required input is the output of an instruction that hasn't completed

Pipelining Questions (courtesy David Jacobs)

- Suppose you've designed a MIPS processor implementation in which the stages take the following lengths of time: IF=20ns, ID=10ns, EX=20ns, MEM=35ns, WB=10ns. What is the minimum clock period for which your processor functions properly? Where should the bulk of your R&D budget go for the next generation of processors?
- Your friend tells you that his processor design is 10x better than yours, since it has 50 pipeline stages to your 5. Is he right? (This is intentionally vague)