ESE532: System-on-a-Chip Architecture

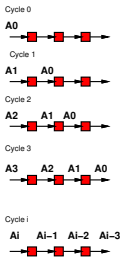
Day 7: September 24, 2018
Pipelining -- Supplement

Register

- On each cycle, the output of the register is the value of the input of the register on the previous cycle.

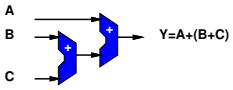
Shift Register

- Illustrate operation
- Data moves through registers
- Shifts to right with each cycle
- Steady state, output at right is value input 3 cycles prior



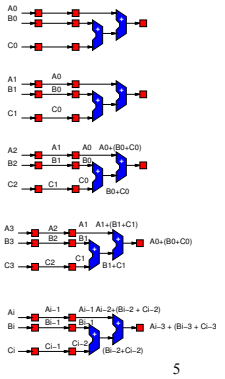
Addition

- Combinational Addition
- Two Adders
- Output is sum of 3 values



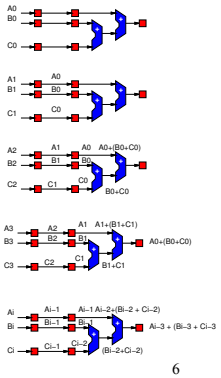
Pipelined Addition

- Here we add registers to compute the addition in part of a pipeline
- Like the shift register values move through the registers over cycles
- The output is the sum of the inputs 3 cycles prior



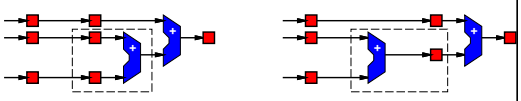
Pipelined Addition

- Note that the output is all inputs from the **same** cycle
- All the subscripts to A, B, C are the same.

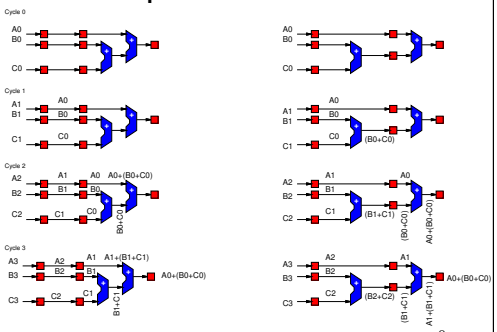


Retimed Add

- Now, we retime the add by applying a lag to the first adder
  - Move registers from its inputs to outputs

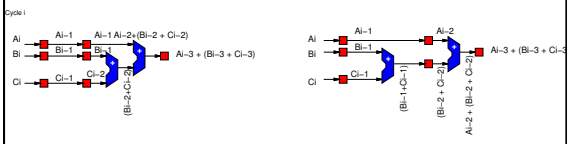


Compare Behavior



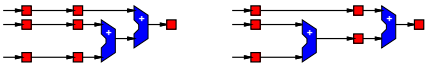
Retimed Behavior

- To external behavior, the output is unchanged after the lag
  - Circuits behave identically



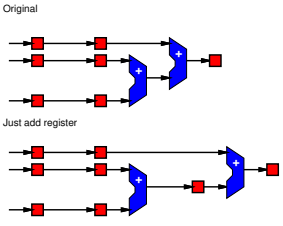
Retimed Circuit Cycle Time

- Original circuit had two adder delays between registers
  - Minimum cycle time = 2*Tadd
  - Throughput = 1/(2*Tadd)
- Retimed circuit has one adder delay between registers
  - Minimum cycle time = Tadd
  - Throughput = 1/Tadd
- By performing the retiming, we doubled the throughput



What-if just added a register?

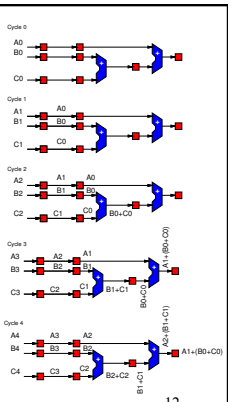
- Rather than lag, what if we just added a register?



Just Add Register Behavior

Note how this is different from the original (and from the retimed)

Here, final output mixes A1 with B0 and C0



Just Add Register Function

Output takes the B and C from one input and adds it to the A that enters one cycle later.