EECS 151/251A Homework 9

Instructor: Prof. John Wawrzynek, TAs: Christopher Yarp, Arya Reais-Parsi

Due Monday, Apr 22nd, 2019

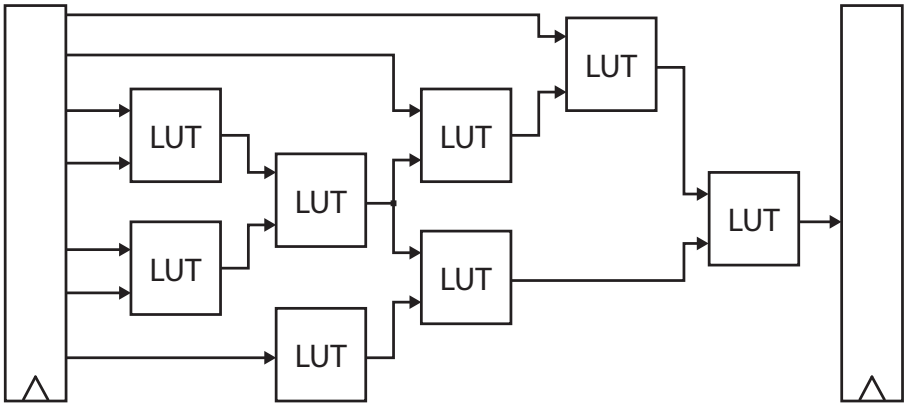
Problem 1: Pipelining for Speed

Given the circuit shown below with $\tau_{clk-q} = \tau_{setup} = 50ps$, and $\tau_{LUT} = 100ps$.

(a) Using pipelining, find the number of stages where the clock frequency is at least 2 times the original (unpipelined) clock frequency.

(b) Find the number of stages where the clock frequency is at least 3x the original.

Assume that LUTs are indivisible (cannot be internally pipelined). For each question above, draw dashed lines in the circuit diagram to indicate where to add registers.



Problem 2: Pipelining in the Presence of Feedback

For the following streaming computation, draw a circuit that uses inter-operator pipelining to maximize the clock frequency.

$y_i = ay_{i-1} + ay_{i-2} + b_0x_i + b_1x_{i-1} + b_2x_{i-2}$

Problem 3: Pipelining in the Presence of Feedback

Consider the design of a datapath to implement the following streaming computation: $y_i = \alpha x_i + y_{i-1}$.

Version: 1 - 2019-04-12 20:23:10-07:00

(a) Draw the diagram of a circuit that has a throughput of 1 result per cycle. Use inter-operator pipeline register(s) to maximize clock frequency.

(b) Draw the diagram of a circuit that has a throughput of 8 results per cycle and operates at a high frequency. *Hint: unroll the loop and use rearrangement of the operators to maximize the clock frequency.*

Problem 4: Simultaneous Multithreading (SMT)

The c-slowing technique is used to allow independent computations to share a single pipeline. When applied to CPU design, this is called "multi-threading" because multiple instruction streams share a single pipeline. Because dependent instructions are spread out in the pipeline, this approach has the beneficial property that with a sufficient number of instruction streams, hazards are eliminated, without the need for special bypassing hardware for data hazards and branch prediction for control hazards. Based on our 3-stage RISC-V pipelined processor, how many threads would be needed to eliminate the data and control hazards? Explain what changes would need to be made to the processor to support this style of execution?

Problem 5: Carry Select Adder

Consider the design of a 48-bit carry select adder. What set of block sizes would lead to minimal delay. Assume that $\tau_{FA} = \tau_{MUX}$.

Problem 6: 16 Port Adder

Based on full-adder cells, draw a circuit that can be used to add a set of 16 1-bit inputs and output the sum.

Minimize the amount of logic required for your implementation.

Problem 7: Carry Look-ahead Adder

For the CLA adder shown in slide 24 of lecture 20, what is total number of gates in series on the critical path, assuming 2-input gates. (It's okay to count AND and OR each as only one gate).

Problem 8: Brent-Kung and Kogge-Stone Adders

For an 8-bit version of the Brent-Kung adder architecture, and considering it's implementation using 2-input gates:

(a) What is the total number of gates?

(b) How many gates are in series on the critical path?

Repeat for the Kogge-Stone adder architecture.