The UNIVERSITY *of* NORTH CAROLINA *at* CHAPEL HILL

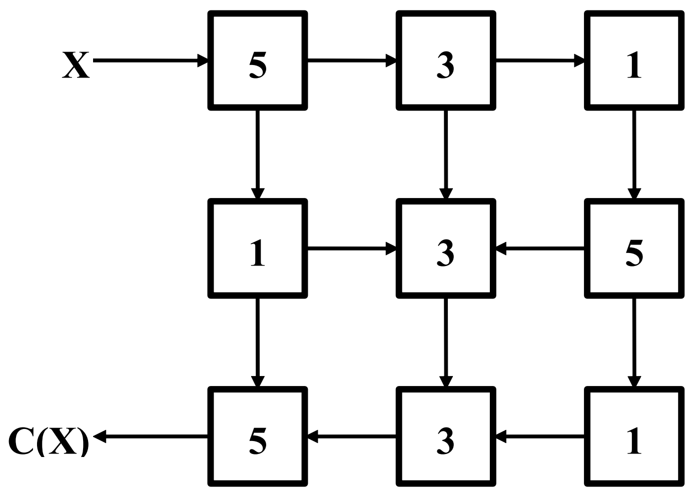
Comp 120 Computer Organization
Spring 2005

Problem Set #5
Issued Thursday,24/2/05; Due Thursday, 3/3/05

Homework Information: Some of the problems are probably too long to be done the night before the due date, so plan accordingly. Late homework will not be accepted. Feel free to get help from others, but the work you hand in should be your own.

Problem 1. Peculiar Pipelines

Peculiar Peripherals, Inc. Builds a combinational encryption device constructed of nine modules as follows:



The device takes an integer value X and computes an encrypted version C(X). In the diagram above each combinational component is marked with its propagation delay in microseconds; contamination delays are zero for each component.

(A) What is the latency and throughput of the combinational encryption device?

(B) Redraw the diagram marking the locations for ideal (zero-delay) registers that will pipeline the device for maximal throughput. Ensure a register at the output and use the *minimum* number of registers necessary.

(C) Give the latency and throughput of your pipelined version. Again assume ideal registers.

Problem 2. How Many Ones could a Turing Machine write if a Turing Machine could write ones?

In the following problems consider a Turing machines with the following specifications. Each Turing machine has *n* states labeled {S1, S2, … S*n*}, the Turing machine begins in state S1and halts by transitioning to the special state S0, each cell of the Turing machine's tape can contain either a "1" or a "0", and each move of the Turing machine based solely on its current state and the value of the tape cell under the tape head. A move consists of first modifying the contents of the current tape cell under the head of the Turing machine, moving the tape either left, L, or right, R, followed by a transition to the next state.

The following truth table defines the behavior of a Turing machine. Note that in this FSM the outputs are a function of both the current state *and* the tape value.

Current State	Tape Value	Write Tape	Move	Next State
S1	0	1	L	S2
S1	1	1	R	S3
S2	0	1	R	S1
S2	1	1	L	S2
S3	0	1	R	S2
S3	1	1	R	S0

(A) How large a ROM is required to implement an *n*-state Turing machine that adheres to the given specification (give the number of words and the bits-per-word)? Ignoring the possibility of equivalent machines, in general how many n-state Turing machines are there?

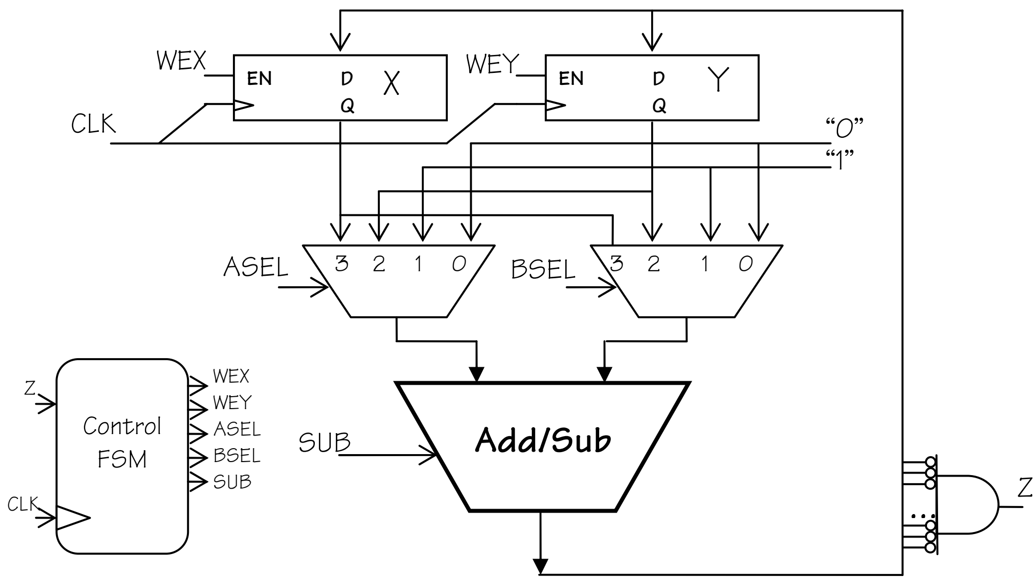
(B) Given an infinite tape (in both directions), with "0"s in every cell. What will the Turing machine described by the truth table above leave on the tape when it halts? Show the status of the tape in the region around the head after each move (just show the first 13 transitions).

(C) Design a 2-state Turing Machine that writes as many "1"s as possible onto an all zero (in both directions) tape and then halts. Hint: you can write four "1"s.

(D) Suppose that we choose to ignore the value of the cell under the read head, thus turning our Turing machine into a finite-state machine. In this case, how many "1"s can we write if the FSM has n states?

Problem 3. "Beating a Path"

You are given the following 32-bit data path and programmable control FSM:



(A) Write a truth table for the control state machine that computes the function $f(x,y) = x * 2^y$.

(B) Write a truth table for the control state machine that computes the function $f(x) = x^2$.
(Hint: $\sum_{i=1}^{n} i = \frac{n(n+1)}{2}$)

(C) Write a truth table for the control state machine that swaps the contents of the X and Y registers.