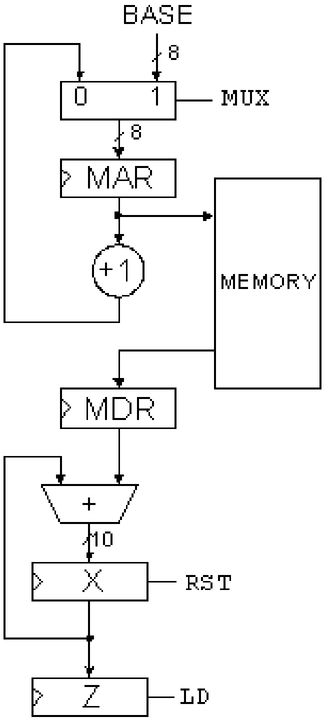
University of California at Berkeley
College of Engineering
Department of Electrical Engineering and Computer Science

EECS150, Spring 2010

Homework Assignment 13: RTL and Scheduling
Due April 30^{th}, 2pm

Homework submission will only be through SVN. Email submissions will not be accepted! Please format your homework as plain text with either PNG or PDF for any necessary figures. Microsoft Visio is installed on the machines in 125 Cory, and is a useful tool for drawing figures of all kinds.

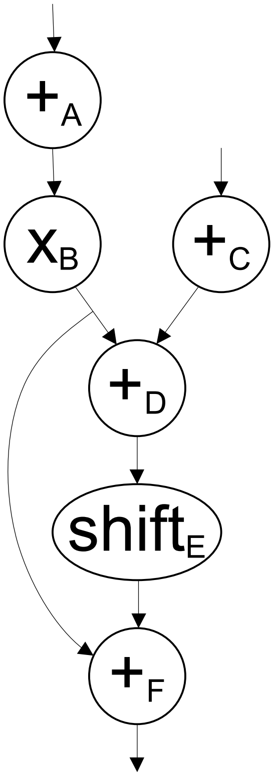
1. Consider the design of a simple processor used to add the contents of blocks of 4 bytes in consecutive memory locations. The datapath circuit for the processor is shown below.



The processor has one data input (8-bit wide) named **BASE**, an input control signal named **ENABLE**, and 3 internal control signals - **MUX**, **LD**, and **RST**. The datapath contains three data registers - **MAR**, **MDR**, and **X**. After the processor performs its operation, the **Z** register is left with the sum of memory locations **BASE**, **BASE + 1**, **BASE + 2**, and **BASE + 3**. We assume that a controller (not shown) will take as input the **ENABLE** signal and generate **MUX**, **RST**, and **LD**. To begin the addition operation, an external circuit asserts **ENABLE** for 1 clock cycle then lowers it for a minimum of 12 cycles.

Write the RTL level description for the sequence of transwers that must occur after the **ENABLE** signal is asserted. Try to minimize the total number of cycles.

2. Imagine a datapath that has four computation units; two adders, a multiplier, and a shifter. Each unit requires an entire clock cycle (minus flip-flop overheads) to complete its operation and is followed by a register to hold its output. The graph in below represents an iterative operation to be completed on the datapath. Each node is labeled with the name of the computation unit that it requires plus a unique letter identifying the node. Note that there is no feedback (or loop carry dependence) in this computation.



Use modulo scheduling to show how to complete four iterations of the loop in the minimum number of cycles. Show your work. Then, fill in the chart shown below the unique integer node numbers from the graph. Use subscripts (1, 2, 3, and 4) to indicate the iteration number. For instance, "C_2" indicates node C of iteration 2.

adder1																			
adder2																			
mult																			
shift																			
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19

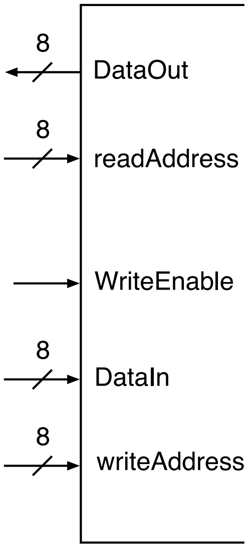
clock cycle

3. Consider the design of a special processor connected to a dual-ported memory, shown below (next page). In the memory is stored an **array** of 8-bit integers, starting at address 0. When started, the processor begins at location 0 and moves through memory forming the sum of all the integers up to that point, storing the sum in each memory location as it goes. The process continues for the entire array.

   An input signal call **`START`** is used to start the process, and an input called **`ENDADDR`** is used to specify the address of the final element in the array.

   Write the RTL description of the processor operation and draw the design of the processor datapath.

   - Use only the following circuit elements:
     - binary adder(s) of any width,
     - register(s) with reset and load-enable,
     - equal-comparator(s),
     - and the memory show below. The memory has *asynchronous read* and *synchronous write* operations.
   - In your design, minimize the processor cycle time and the number of cycles in the inner-loop.
   - Remember to use a comma, ",", to separate RTL operations that occur on the same clock cycle, and the ";" to seperate operations on different cycles.


8
DataOut
8
readAddress
WriteEnable
8
DataIn
8
writeAddress