Kurtis Reid

**Review Ch.3-4**

**Arithmetic and Datapath**

1. Describe the algorithm that performs the multiplication of two binary integers.

**Test multiplier, if multiplier = 1 add mtltupilicand to product and place the result in product register, then shift mutiplican reg left by 1 bit, shift multipyer reg right by 1 bit, check if on 32 repition, if on 32 repition then finish, else repeat.**

1. Perform the division of 100111 by 100.

111 R11

1. What is the difference between single and double precision?

Single 32 bits

- smallest value = -126

- largest value = 254

Double 64 bits

- smallest value = -102

- largest value = 1023

1. Use the IEEE 754 standard for the representation of the following numbers:
   1. -35.25

1011110100001101

* 1. 125.75

0 01111001111101110

1. What is the overflow? What is the underflow? Give an example of overflow and underflow. When does it occur?

Overflow- result is too large to represent in 32 bits

11001 + 11111 = overflow

Underflow- result of a floating point is smaller in magnitude (closer to zero) than the smallest value representable

result with value less then 2^-127

1. Perform the addition of these numbers:

1.110001 \* 2-7 + 1.0001 \* 25 + 1.11110 \* 21 =

1.001011110001110001x2^5

1. Perform the multiplication of these numbers:

1.110001 \* 2-4 +1.110 \* 21 =

11.000100111\*2^-3

~~1.1100111000x2^1~~

1. ~~(Skip this question) What is the benefit of using the guard and round in floating point arithmetic?~~
2. What is the job of a multiplexor? What is the value assigned to the data selector of a multiplexor if we want to output the signal carried by the fourth wire entering in input the multiplexor?

**Multiplexor selects an input line to send in output. The selected input line depends on the value of the data selector line. When the value of the data selector line is 11, which = 3, the value of the fourth input line will be sent in output.**

1. Consider the simple single-cycle datapath described in your book. How many functional units (or datapath elements if you prefer) are part of the implementation of the datapath?

**The instruction memory,**

**the register,**

**the ALU,**

**the data Memory,**

**and multiple adders**

1. What is the purpose of the Data Memory in the single-cycle implementation? **Load/Store**
2. A state element is also called a …………… element. **Memory element**
3. What is an edge-trigger clocking? **A clocking scheme in which all state chnages occur on a clock edge**
4. What is the need for a clock? **To ensure predictability and define when signals can be read and when they can be written.**
5. Describe the path taken by the load instruction in a single-cycle implementation scheme. (answer the same question for all the other types of instructions)

Insruction fetch:

Get adress from Instruction memory

add 4 to pc

ID:

* + 1. Registers; Read data 1 & 2, sign extend

EX: Data 1 to ALU, data 2 & sign extend to multiplexor, sign extend to shift left 2 to add result, result from multipexor to ALU, zero and ALU result to data memory

MEM: Addresult to multiplexor, Data memory to multiplexor

WB: Multiplexor to write register in registers

1. Which instructions use the Sign Extend unit in the single-cycle implementation scheme? **All the I format instructions**
2. When is the MemtoReg asserted in a single-cycle implementation scheme? Give the name of an instruction that requires the assertion of the MemtoReg.

**The memtoReg control is asserted to 1 when the instruction wants to write the result retrived from the Data memory back to the register. The load instruction requires the assertion of MemtoReg.**

1. Explain the purpose of the control unit in a single-cycle implementation scheme. **To signal what values to use based on the instructions**
2. There are two possible 5bit chunks, i.e. [20-16] and [15-11] of the 32 bit instructions that are used to indicate the Write Register. Describe in which case the first is used and in which case the second is used.

**When an Rformat is used, the 15-11 bit positions are used and control line RegDest is set to 1 to get from the Mux the apropiate destination register. When an I format instruction is used, the 20-16 bit positions are used.**

1. Why is a multi-cycle implementation scheme introduced? **To improve performance**
2. What are the differences of the multi-cycle implementation scheme with respect to the single-cycle implementation scheme?

**Multi-cycle allows multiple instructions to be implimente while at the same time, having the resouces be in constant use. Single cycle impliments one instruction then impliments the next when the second has finished.**

1. The time to execute a load in a multi-cycle implementation scheme is slower than the time to execute a load in a signle-cycle implementation scheme. Why is that? **The load instruction is the longest instruction in both cycle implimentations. In multi-cycle, in order to exicute in a proper way the load instruction, information must be carried between stages in registers that must be written at the end of each stage. This additional activity adds an overload that slows the overall time of the load instruction in the muti-cycle.**
2. Trace a store instruction on the multicycle implementation scheme. What are the values of the control signals in each of the cycles?
3. In which stage of the multicycle implementation, the following steps are performed?
   1. the Instruction[25:21]of the instruction is read and given in input to the ALU
   2. the Instruction[20:16]of the instruction is read and given in input to the ALU
   3. the PC and the (sign-extend(Instruction[15-0] << 2) is given in input to an adder;

**In the EX stage of a branch instruction. The first two steps read exicute the adress calculation of a branch.**

1. Which is the shortest instruction in a multi-cycle implementation scheme?

**Jump**

1. ~~(Skip this question) Is it possible that the control signal PCWriteCond can be replaced by the PCSource[0] in the multi-cycle implementation scheme. If so, when?~~
2. What is a “don’t care” symbol and when it is used? This answer is in the book not in the slides.

**The symbol X. It is usd to indicat when the value of a control line is independent from the result of the executing instruction.**

1. Consider the following code:

Repeat:lw $t1, 0($t2)

subi $t1, $t1, 3

sw $t1, 0($t2)

addi $t3, $t3, 12

bne $zero, $t3, Repeat

and simulate it on a 5-stages pipeline.

Lw $t1, 0($t2)

1. What types of hazards are available in a pipelined implementation? Describe each type. **Structure hazards- required resource is busy. Data hazard- need to wait for previous instruction to finish. Control hazard – descion on control action depends on previous action**
2. Consider the following code.
   1. How many cycles will be required to properly execute the following code if NO additional techniques are used to prevent or eliminate stalling? Specify also where each stall occurs.

add $t1, $t2, $t3

subi $t1, $t1, 3

lw $t1, 0($t2)

add $t3, $t3, $t1

bne $zero, $t3, Loop **15**

1. How many cycles will be required to properly execute the previous code if forwarding or code optimization is used to minimize stalling? Show where forwarding, stalling, or instruction permutation occurred. **10 cycles**
2. What is forwarding and how is it realized? **Use result when it is computed, realized by making extra connections to data path**
3. Can forwarding eliminate any data hazard? Explain. **No, because the information may not be avaliable when it is needed.**
4. What is dynamic branch prediction and where is it used? **Tries to prevent data hazards in deeper and superscalar pipelines**
5. How are exception handled? **In System Control Coprocessor, save PC of offending instruction, save indication of program, jump to handler**