CDA 4205 Computer Architecture

Assignment 5: Single-Cycle Processor Implementation

1. (15 pts) Describe the effect that a single stuck-at-0 fault (i.e., the signal is always 0 regardless of what it should be) would have for the signals shown below, in the single-cycle Datapath. Which instructions, if any, will not work correctly? Explain why.

Consider each of the following faults separately:

1. RegWrite = 0

**No register can be written in the register file, and instructions that require writing to a register will not work.**

1. RegDst = 0

**Instructions will only write to Rt instead of RT because RD can never be used**

1. ALUScr = 0

**Instructions that use immediates can not be used because the second input will always come from the register file.**

1. MemtoReg = 0

**Load instructions will not work, because only values from the ALU will be written to the register**

1. Branch = 0

**Branch instructions will not work, because there will be no indication of a branch instruction.**

1. (15 pts) Repeat question 1 but this time consider stuck-at-1 faults (the signal is always 1).

**All instructions will attempt to write a value into the register file, this will disrupt instruction meant to load the data, as it will change it before it’s loaded.**

**Instructions will only write to Rd instead of Rt because RT can never be used**

**The second input will be stuck to whatever the “immediate” value is, instead of itt coming from the register. Instructions that use register value will not work**

**Only values frfom the memory will be written down, even when it’s supposed to be the ALU value.**

**Only branch instructions will work, because there will be a constant indication of a branch instruction.**

1. (15 pts) We wish to add the instruction **jalr** (jump and link register) to the single-cycle datapath. Add any necessary datapath and control signals and draw the result datapath. Show the values of the control signals to control the execution of the **jalr** instruction.

The jump and link register instruction is described below:

**jalr rd, rs # rd = pc + 4 , pc = rs**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **op6 = 0** | **rs5** | **0** | **rd5** | **0** | **Funct6 = 0x9** |

**I feel like this would take a horrible amount of time to draw out…**

1. (15 pts) Suppose we add the multiply and divide instructions. The operation times are as follows:

Instruction memory access time = 190 ps, Data memory access time = 190 ps,

Register file read access time = 150 ps, Register file write access = 150 ps

ALU delay for basic instructions = 190 ps, ALU delay for multiply or divide = 550 ps

Ignore the other delays in the multiplexers, control unit, sign-extension, etc.

Assume the following instruction mix: 30% ALU, 15% multiply & divide, 20% load, 10% store, 15% branch, and 10% jump.

1. What is the total delay for each instruction class and the clock cycle for the single-cycle CPU design?

**ALU – 190 + 150 + 190 + 150 = 680ps**

**Multiply & Divide – 190 + 150 + 550 + 150 = 1040ps**

**Load – 190 + 150 + 190 + 190 + 150 = 870ps**

**Store – 190+ 150 + 190 + 190 = 720ps**

**Branch – 190 + 150 + 190 = 530ps**

**Jump = 190 + 150 = 340**

**Max delay is 1040ps**

1. Assume we fix the clock cycle to 200 ps for a multi-cycle CPU, what is the CPI for each instruction class and the speedup over a fixed-length clock cycle?

**ALU – 4 cycles**

**Multiply & Divide – 6 cycles**

**Load – 5 cycles**

**Store – 4 cycles**

**Branch – 3 cycles**

**Jump – 2 cycles**

**Average – (4 \* 30%) + (6 \* 15%) + (5 \* 20%) + (4\*10%) + (3 \* 15%) + (2 \* 10%) = 4.15**

**Speedup – (1040)/(200\*4.15) = 1.25 speedup**

* **Submission Requirements**
* Your solutions must be in a single file with a file name yourname-hw6.
* If scanned from hand-written copies, then the writing must be legible, or loss of credits may occur.
* Only submissions via the link on Canvas where this description is downloaded are graded. Submissions to any other locations on Canvas will be ignored.
* Late submissions are accepted for a maximum of 3 late days with 20% assignment credit off as late penalization. Assignments submitted after 3 late days will not be accepted.