Dylan Thornburg

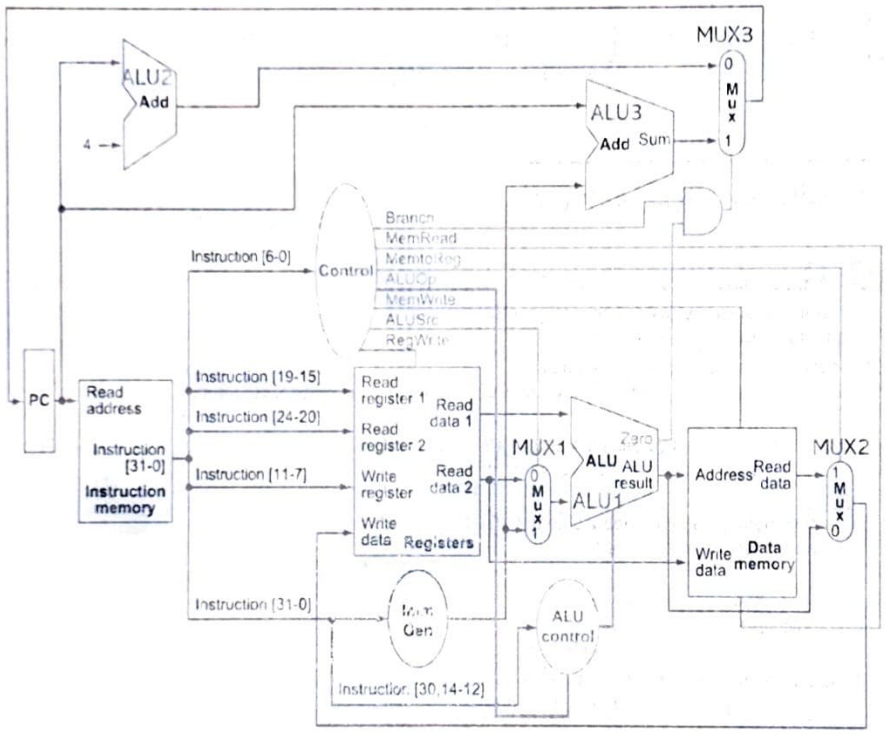
SANTA CLARA UNIVERSITY	ECEN 122 Winter 2024	Hoeseok Yang

Homework #5: RISC-V datapath

Answer the following problems and upload your answers to Camino in pdf format.

Posted: February 15, 2024, Due by: 9pm on February 21, 2024

1. [1 pts] Below is the simple control and datapath that supports RISC-V R-type, load/store, and control branch instructions.



The effects of some control signals (output of *Control*) are summarized in the following table.

Signal name	Effect when deasserted	Effect when asserted
RegWrite	None.	The register on the Write register input is written with the value on the Write data input.
ALUSrc	The second ALU operand comes from the second register file output (Read data 2).	The second ALU operand is the sign-extended, 12 bits of the instruction.
PCSrc	The PC is replaced by the output of the adder that computes the value of PC + 4.	The PC is replaced by the output of the adder that computes the branch target.
MemRead	None.	Data memory contents designated by the address input are put on the Read data output.
MemWrite	None.	Data memory contents designated by the address input are replaced by the value on the Write data input.
MemtoReg	The value fed to the register Write data input comes from the ALU.	The value fed to the register Write data input comes from the data memory.

Name:_____ Student ID:_____

A. (0.2 pts) What are the values of the following control signals for an R-type instruction add? Each should be either of 0, 1, or don't care (X).

RegWrite	ALUSrc	PCSrc	MemRead	MemWrite	MemtoReg
1	0	0	0	0	0

B. (0.2 pts) What are the values of the following control signals for an S-type instruction sw (store word)? Each should be either of 0, 1, or don't care (X).

RegWrite	ALUSrc	PCSrc	MemRead	MemWrite	MemtoReg
0	1	0	0	1	X

C. (0.2 pts) State one RISC-V instruction example that has the selection signal asserted for both MUX1 and MUX2.

lw x7, 0(x5)

D. (0.1 pts) We have three ALUs in the above datapath design: ALU1, ALU2, and ALU3. For control branch instructions (e.g., beq or bne), which one is used for evaluating the branch condition?

ALU1

E. (0.1 pts) We have three ALUs in the above datapath design: ALU1, ALU2, and ALU3. In MIPS control branch instructions (e.g., beq or bne), which one is used for calculating the target address?

ALU3

F. (0.2 pts) State whether the above two ALUs (answers to D and E) can be executed at the same time or not. In other words, does one ALU need to wait for the completion of the other?

ALU3 needs to wait for ALU 1. Not executed at the same time

Name:_____ Student ID:_____

2. [0.6 pts] The result of a benchmark running on a CPU (clock rate = 4GHz) has an instruction count of 2.4×10^{12} and an execution time of 720s. Use the following equation:

CPU time = (Instruction Count) x (Cycles Per Instruction) x (Clock Cycle Time)
= (Instruction Count) x (Cycles Per Instruction) / (Clock Rate)

A. (0.3 pts) Calculate the average number of clock cycles per instruction (CPI) in this case.

$$720 = \frac{2.4 \times 10^{12} X}{4 \times 10^{9}}$$

$X \approx$ 1.2 CPI

B. (0.3 pts) Suppose that we are developing a new generation of this CPU running at 5GHz. We have added some additional instructions to the instruction set in a way that the number of instructions in the same benchmark has been reduced by 20%. When the execution time taken for this benchmark is 600s, calculate the new CPI.

$$600 = \frac{(.8)(2.4 \times 10^{12}) X}{5 \times 10^{9}}$$

$X =$ 1.5625 CPI

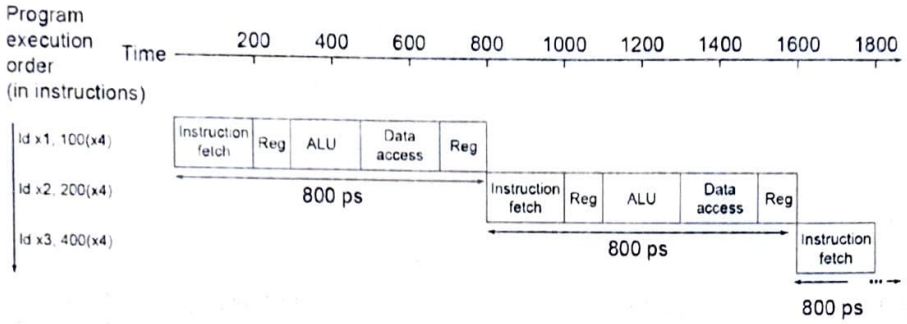
3. [0.4 pts] We generally prefer to have a faster CPU (smaller execution time) on a smaller silicon die (smaller hardware area). Read this wikipedia page (https://en.wikipedia.org/wiki/Dark_silicon) and briefly state (in one sentence or two) the problem we may encounter when implementing a fast CPU (with a high clock frequency) on a small silicon die.

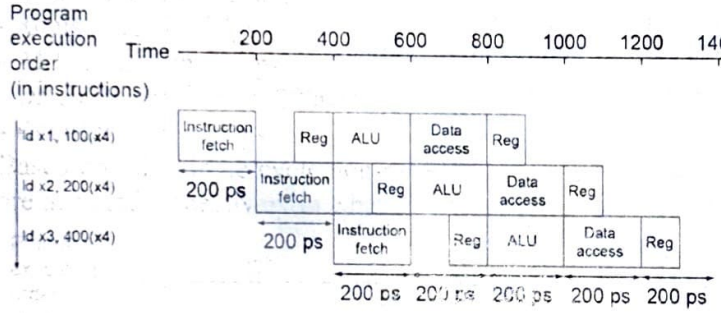
The problem is dark silicon. Essentially, when ICs get so small, transistors on silicon do not increase their efficiency proportionately to their increase in number. This leads to a sharp increase in power density and unsafe temperatures.

Name:_____ Student ID:_____

4. [1 pts] Let us suppose that each hardware component takes the following time:
   - Instruction fetch: 200ps
   - Register read: 100ps
   - ALU operation: 200ps
   - Data memory access: 200ps
   - Register write: 100ps

   Based on this, single-cycle non-pipelined and pipelined RISC-V datapath executions are depicted respectively in what follows:

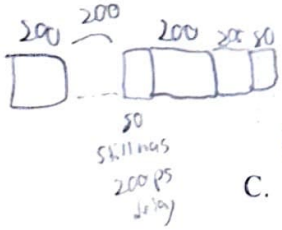




A. (0.3 pts) What are the latencies (response times) of *ld* in the non-pipelined RISC-V datapath and in the pipelined RISC-V datapath, respectively?

non-pipelined: 800ps) one instruction
pipelined: 900ps
3 instr ~ 2400 ps
~ 1300

B. (0.3 pts) Suppose that we reimplement the register file (RF) to enhance the speed and register read/write now takes 80ps. What is the new latency of *ld* in the non-pipelined RISC-V implementation? And, how about the pipelined one?



non-pipelined: 760ps) one instruction
pipelined: 880ps
b/c
3 instr ~ 2280
~ 1280

C. (0.4 pts) In the previous case (register read/write time: 100ps→80ps), does the new RF change the throughput of the pipelined datapath?

old throughput
800·x ≈ 4 times increase
200·x+900

new
760·x ≈ 3.8 times increase
200·x+880

it is changed, and it makes sense b/c the non pipelined format got a better decrease (40 ps vs 20 ps per cycle).

Name:_____ Student ID:_____