CS 3330 Computer Architecture, Spring 2020
HW 3: Dependency, SIMD, Systolic Arrays

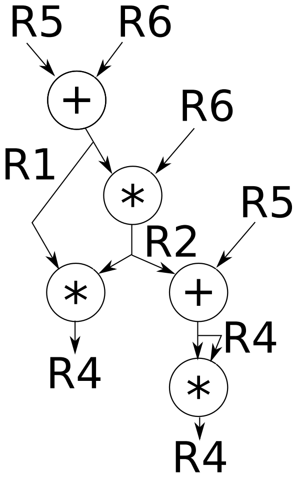
Instructor: Prof. Samira Khan
TAs: Amel Fatima, Sihang Liu, Korakit Seemakhupt, Yasas Senevirathne, Yizhou Wei
Min Jae Lee, Nikita Semichev, Yuying Zhang.

Assigned: Feb 24, 2020
Due: **March 6, 2020, plus 4-days of auto-extension for everyone**

1 Stalling and Data Forwarding [40 points]

In this problem, we provide you a data flow graph. It is executed in a machine that can process instructions in the data-flow order. The machine in this problem has the following characteristics:

- The processor is fully pipelined with four stages: Fetch, decode, execute, and writeback.
- For all instructions, fetch takes 1 cycle, decode takes 1 cycle, and writeback takes 1 cycle.
- The processor implements `ADD` and `MUL` instructions only. Both the adder and multiplier are fully pipelined. `ADD` instructions take 3 cycles and `MUL` instructions take 4 cycles in the execute stage. Note that the adder and multiplier can broadcast results in the same cycle.
- Given that the processor has one separate adder and one separate multiplier, two independent `ADD` and `MUL` instructions can enter the execute stage if there is no other dependency.
- The processor has data forwarding available after the execution stage. It means that output of an ADD can be forwarded to the next dependent instruction after the execution stage.



(a) Now, use the data flow graph to fill in the table below with the five instructions being executed on the processor. The source registers for the first instruction can be specified in either order. Give instructions in the following format: "opcode, destination, source1, source2."

OP	Dest	Src 1	Src 2

(b) Now show the full pipeline timing diagram below for the sequence of five instructions that you determined above, from the fetch of the first instruction to the writeback of the last instruction. Assume that the machine stops fetching instructions after the fifth instruction.

As we saw in class, use "F" for fetch, "D" for decode, "En" to signify the nth cycle of execution for an instruction, and "W" to signify writeback. You may or may not need all columns shown. For example, an add instruction will look like this if there is no stall: F | D | E1 | E2 | E3 | W

Cycle:	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
Instruction 1																			
Instruction 2																			
Instruction 3																			
Instruction 4																			
Instruction 5																			

(c) How many cycles it will take to process all instructions?

2 Vector Processing [40 points]

Consider the following piece of code:

```
for (i = 0; i < 100; i ++)
  A[i] = ((B[i] * C[i]) + D[i])/2;
```

(a) This code is translated into assembly language using the following instructions in the ISA (note the number of cycles each instruction takes is shown next to each instruction):

Opcode	Operands	Number of Cycles	Description
LEA	Ri, X	1	Ri ← address of X
LD	Ri, Rj, Rk	11	Ri ← MEM[Rj + Rk]
ST	Ri, Rj, Rk	11	MEM[Rj + Rk] ← Ri
MOVI	Ri, Imm	1	Ri ← Imm
MUL	Ri, Rj, Rk	6	Ri ← Rj x Rk
ADD	Ri, Rj, Rk	4	Ri ← Rj + Rk
ADDI	Ri, Rj, Imm	4	Ri ← Rj + Imm
RSHFA	Ri, Rj, amount	1	Ri ← RSHFA (Rj, amount)
BRcc	X	1	Branch to X based on condition codes

```
MOVI    R1, 99  // 1 cycle
LEA     R0, A // 1 cycle
LEA     R2, B // 1 cycle
LEA     R3, C // 1 cycle
LEA     R4, D // 1 cycle
LOOP:
LD      R5, R2, R1 // 11 cycles
LD      R6, R3, R1 // 11 cycles
MUL     R7, R5, R6 // 6 cycles
LD      R5, R4, R1 // 11 cycles
ADD     R8, R7, R5 // 4 cycles
RSHFA   R9, R8, 1  // 1 cycle
ST      R9, R0, R1 // 11 cycles
ADD     R1, R1, -1 // 4 cycles
BRGEZ   R1  LOOP   // 1 cycle
```

How many cycles does it take to execute the program?

(b) Here is the Cray-like vector assembly code to perform this operation in the shortest time possible. Assume that there are 8 vector registers and the length of each vector register is 50.

Opcode	Operands	Number of Cycles	Description
LD	Vst, n	1	Vst ← n (Vst = Vector Stride Register)
LD	Vln, n	1	Vln ← n (Vln = Vector Length Register)
VLD	Vi, X	11, pipelined	
VST	Vi, X	11, pipelined	
Vmul	Vi, Vj, Vk	6, pipelined	
Vadd	Vi, Vj, Vk	4, pipelined	
Vrshfa	Vi, Vj, amount	1	

```
LD     Vln, 50
LD     Vst, 1
VLD    V1, B
VLD    V2, C
VMUL   V4, V1, V2
VLD    V3, D
VADD   V6,  V4, V3
VRSHFA V7,  V6, 1
VST    V7,  A


VLD V1, B+50
VLD V2, C+50
VMUL V4, V1, V2
VLD V3, D+50
VADD V6, V4, V3
VRSHFA V7, V6, 1
VST V7, A+50
```

How many cycles does it take to execute the program on the following processors? Assume that memory is 16-way interleaved.

(i) Vector processor without chaining, 1 port to memory (1 load or store per cycle).

(ii) Vector processor with chaining (forwarding), 1 port to memory.

3 BONUS: Systolic Array and TPU [20 points]

Read these two papers and write a short summary of the papers. Please follow these slides on how to critically analyze papers. https://www.cs.virginia.edu/~smk9u/CS3330S20/reviewing.pptx

(a) Kung, H. T., ”Why Systolic Architectures?”, IEEE Computer 1982. https://www.cs.virginia.edu/~smk9u/CS3330S20/kung_-_1982_-_why_systolic_architectures.pdf

(b) Jouppi et al., ”In-Datacenter Performance Analysis of a Tensor Processing Unit”, ISCA 2017. https://dl.acm.org/citation.cfm?id=3080246

4 Handin

You should electronically hand in your homework (in pdf format) to Collab.