Homework 1 (due September 10, 2019)

Question 1: When parallelizing an application, the speedup is determined by the percentage of the application that can be parallelized and the additional cost of communication between parallel processes. Amdahl's law takes into account the formal but not the latter.

a) What is the speedup with 8 processors if 80% of the application is parallelizable and for every processor added, the communication overhead is 0.5% of the original execution time?
b) Generalize the speedup formula from part (a) to an arbitrary number of processors, N, and determine the number of processors that will result in the highest speedup in the application.
c) Repeat part (b) assuming that the fraction of the application that is parallelizable is F rather than 0.8.

a)

$$SpeedUp = \frac{1}{(1-F) + \left(\frac{F}{P}\right) + ((P-1) * C)}$$

$$SpeedUp = \frac{1}{(1-0.8) + \left(\frac{0.8}{8}\right) + ((8-1) * 0.005)}$$

$$SpeedUp = 2.99$$

b)

$$SpeedUp = \frac{1}{0.2 + \left(\frac{0.8}{P}\right) + ((P-1) * 0.005)}$$

$$\frac{d}{dP}\left(\frac{1}{0.2 + \frac{0.8}{P} + (P-1) \times 0.005}\right) = \frac{32000 - 200\,P^2}{(P^2 + 39\,P + 160)^2}$$

To find the max, we set the derivative to 0 and then find the value for P:

$$\frac{32000 - 200\,P^2}{(P^2 + 39\,P + 160)^2} = 0$$

$$P \approx 12.6491$$

Therefore, either 12 or 13 processors will result in the highest speedup in the application. Trying both values shows that the highest speedup is at 12 processors.

c)

$$SpeedUp = \frac{1}{(1-F) + \left(\frac{F}{P}\right) + \left((P-1) * 0.005\right)}$$

$$\frac{d}{dP}\left(\frac{1}{(1 - F) + 0.005\,(-1 + P) + \frac{F}{P}}\right) = \frac{F - 0.005\,P^2}{(F\,(P - 1) + (-0.005\,P - 0.995)P)^2}$$

To find the max, we set the derivative to 0 and then find the value for P:

$$\frac{F - 0.005\,P^2}{(F\,(P - 1) + (-0.005\,P - 0.995)P)^2} = 0$$

$$P^2 = \frac{F}{0.005}$$

$$P = 10\sqrt{2}\sqrt{F}$$

Question 2: The study of usage of high-level language constructs of a computer system suggests that procedure calls are one of the most expensive operations. You are considering an architecture modification to the system that reduces the loads and stores normally associated with procedure calls and returns. Your analysis of the design and workloads reveal the following information:

- The clock rate after applying your modification is 5% slower.
- Thirty percent of the instructions in a typical program (before applying your modification) are loads or stores.
- The modified version executes two-thirds as many loads and stores as the original version. For all other instructions, the dynamic execution counts are unchanged.
- All instructions (including load and store) take one clock cycle.

(a) Will your modification lead to a faster execution? Justify your decision quantitatively.
(b) Derive a general formula to determine if the modification leads to faster execution in terms of the clock rate slowdown, w, the reduction of the number of loads/stores, r, and the percentage of the original instructions that are load/store, F. Note that in part (a), w = 0.05, r = 1/3 and F = 0.3.

a)

$$I_{new} = I_{old} \times (1 - 0.3) + I_{old} \times 0.3 \times \frac{2}{3}$$

$$I_{new} = 0.9 \times I_{old}$$

$$C_{new} = C_{old} \times (1 - 0.05)$$

$$C_{new} = 0.95 \times C_{old}$$

$$SpeedUp = \frac{I_{old}/C_{old}}{I_{new}/C_{new}} = \frac{I_{old}/C_{old}}{0.9 \times I_{old}/0.95 \times C_{old}} = 1.06$$

The modification will lead to faster execution because the speed up is greater than 1.

b)

$$\frac{I_{old}/C_{old}}{(1 - F \times r) \times I_{old}/(1 - w) \times C_{old}} = \frac{1 - w}{1 - F \times r}$$

$$\frac{1 - w}{1 - F \times r} > 1 \quad means\ faster\ execution$$

Question 3 (Variable length op-code): Assuming that you are designing an instruction set with a fixed 32-bit instruction length for a machine with 64 general purpose registers, and that you want to provide three different instruction formats:

- Format 1: op-code R1, R2, R3 ; the instruction requires three register addresses
- Format 2: op-code R1, R2, I(R3) ; the instruction requires three register addresses and a constant I
- Format 3: op-code R2, L(R3) ; the instruction requires two register addresses and a constant, L

(a) Assuming that you want to use 12 bits for the constant I, what is the maximum number of instructions that may be of Format 2?
(b) Given your answer in (a), and assuming that you want to have at least 60 instructions of Format 3, what would be the maximum size of the constant L?
(c) Given your answers in (a) and (b), what is the maximum number of instructions that may be of Format 1?
(d) Discuss the ramification of having 128 registers (instead of 64) on the design of the ISA.

a) Maximum number of instructions for Format 2 = 3
For Format 2:
Number of bits to represent each register = 6
Number of bits to represent R1, R2 and R3 = 18

Number of bits to represent R1, R2, R3 and I = 30
Thus, we have maximum 2 bits left for opcode.
So, we can support at most 2^2 = 4 different op-codes.
Since we need to provide Format 1 (ex. op-code R1, R2, R3) instructions and Format 3 (ex. op-code R2, L(R3)) instructions, we can use one of the op-code (for example, 11) to signal an op-code extension for Format 3 and Format 1 instructions, thus resulting in 3 different Format 2 instructions.

b) Number of bits to represent at least 60 instructions = 6 bits
Number of bits to represent Format 3 type = 2 bits (Given the answer in part a)
Number of bits for opcode of Format 3 = 2 + 6 = 8
Number of bits to represent R2 and R3 = 12 bits
Number of bits available for constant L = 12 bits
Therefore, maximum size of the constant L = 2^12

c) Given the info in part (a) and (b)
We can have 60 instructions for Format 3.
Assuming the starting bits for opcode of Format 3 is 11 and the following 6 bits being the opcode,
The range of 8 bits opcode of Format 3 will be = 11 000000 to 11 111011
Thus, 11 1111XX XXXXXX can be used to represent Format 1 instructions.
Therefore, total number of Format 1 instructions = 2^8 = 256.

d) If we have 128 total general purpose registers, we will require 7 bits instead of 6 to represent each register. Although having more registers is helpful, this will mean the number of bits available for opcode and constants I and L will be much less. Thus, we will not be able to represent enough instructions with this design of the ISA.

Question 4 (Processor design): Consider the processor data-path shown in this figure, which supports the execution of R-type, lw, sw and beq instructions.

(a) Specify the values of the control signals (x1, x2, x3, L1, L4 and L5) for each of the four types of instructions (R-type, lw, sw and beq). Use the signal convention specified in slide 10 of lecture 3)
(b) Modify the datapath to include the jalr instruction which
1) jumps to the address PC+4+Imm, where Imm is a relative address stored in bits 31-12 of the instruction (see slide 21 of lecture 2)
2) stores the value of PC+4 in the register whose address is specified in bite 11-7 of the instruction.

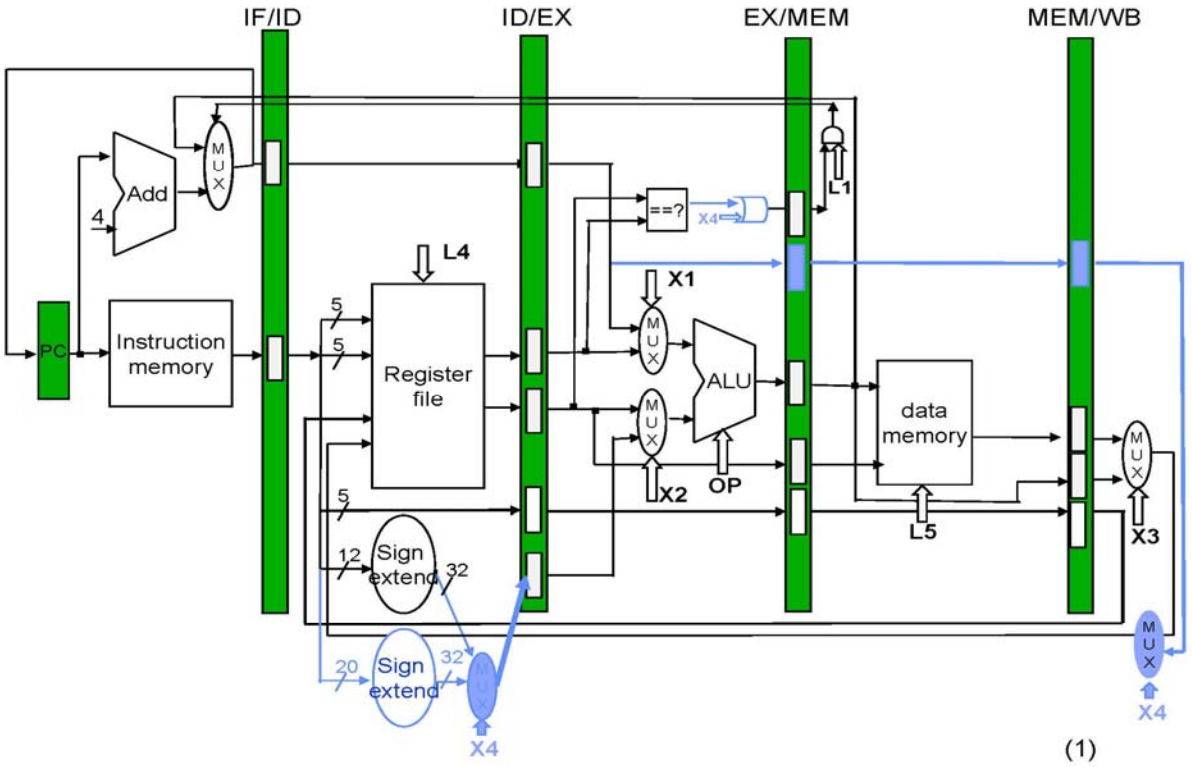
You can either manually draw your changes into this figure or modify the power point slide to incorporate the new paths. You may have to add new components (such as multiplexers) and new control signals.

(c) Specify values of the control signals (the original ones and the new ones introduced in part b) needed to execute the new jalr instruction.

(a)

	R-type	lw	sw	beq
X1	1	1	1	0
X2	0	1	0	1
X3	1	0	1/0	1/0
L1	0	0	0	1
L4	1	1	0	0
L5	00	01	10	00

(b)



(c) X1 = 1
X2 = 1
X3 = 1/0
X4 = 1
L1 = 1
L4 = 1
L5 = 00