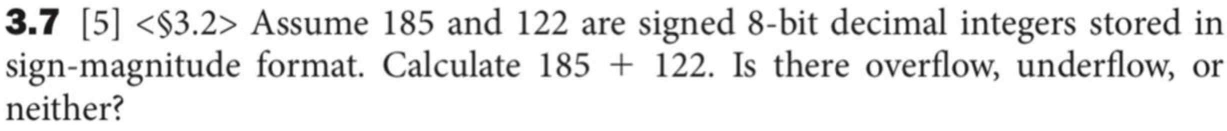
George Suarez

CSE 401

November 07, 2018

Homework 2


3.7 [5] <§3.2> Assume 185 and 122 are signed 8-bit decimal integers stored in sign-magnitude format. Calculate 185 + 122. Is there overflow, underflow, or neither?

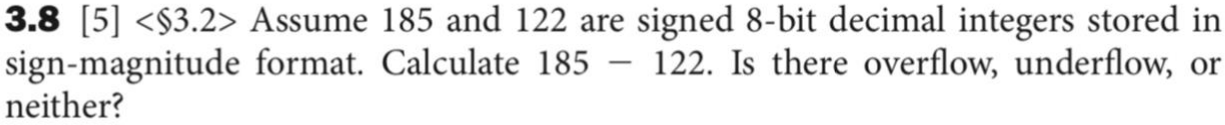
$$(185)_{10} \rightarrow (10111001)_2$$

$$(122)_{10} \rightarrow (01111010)_2$$

Calculating $185 + \ 122$:

$$\begin{array}{r l} & 1\ 0\ 1\ 1\ 1\ 0\ 0\ 1 = (185)_{10} \\ + & 0\ 1\ 1\ 1\ 1\ 0\ 1\ 0 = (122)_{10} \\ \hline & 0\ 1\ 0\ 0\ 1\ 1\ 0\ 0 \text{ with carry of 1} \end{array}$$

Since there is a carry after calculating 185 and 122, then there is an **overflow**.


3.8 [5] <§3.2> Assume 185 and 122 are signed 8-bit decimal integers stored in sign-magnitude format. Calculate 185 − 122. Is there overflow, underflow, or neither?

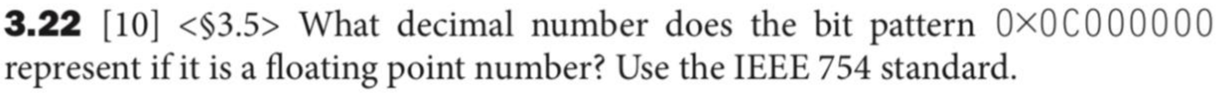
$$(185)_{10} \rightarrow (10111001)_2$$

$$(122)_{10} \rightarrow (01111010)_2$$

Calculating $185 - 122$:

$$\begin{array}{rl} & 1\,0\,1\,1\,1\,0\,0\,1 = (185)_{10} \\ + & 1\,0\,0\,0\,0\,1\,1\,0 = (-122)_{10} \\ \hline & 0\,0\,1\,1\,1\,1\,1\,1 = (69)_{10} \end{array}$$

Since the result ended with a positive result after subtracting a positive number from a negative number, then there is an **overflow**.



3.22 [10] <§3.5> What decimal number does the bit pattern 0×0C000000 represent if it is a floating point number? Use the IEEE 754 standard.

Convert hex to binary:

0	C	0	0	0	0	0	0
0000	1100	0000	0000	0000	0000	0000	0000

Sign Bit (Left most bit): 0

Exponent Bits (8-bits): $(00011000)_2 = (24)_{10}$

Fraction Bits (22-bits): $1.0000 \ldots 000$

Exponent with bias: (using single-precision): $24 - 127 = -103$

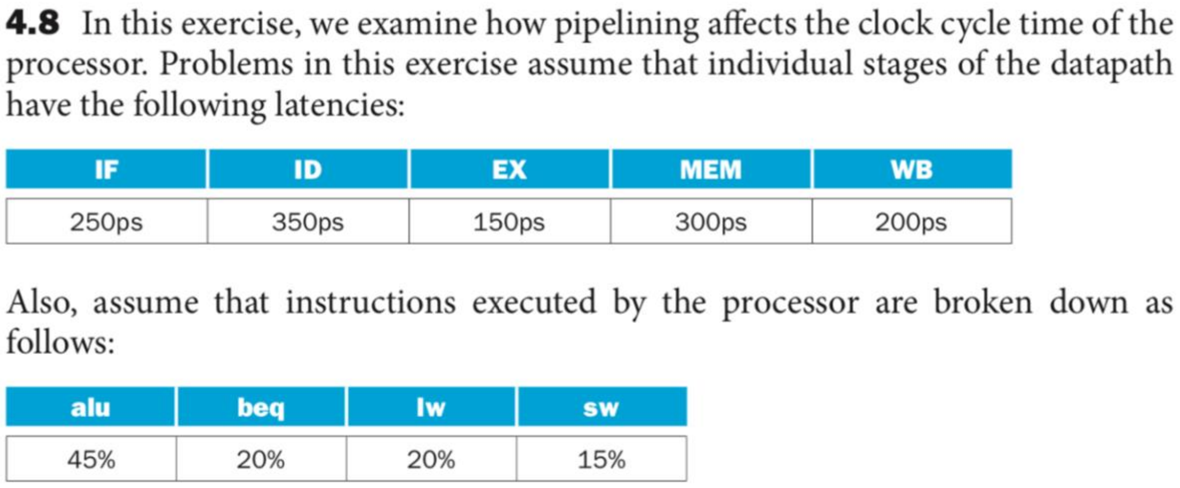
Following the IEEE 754 Standard:

$(-1)^S \times (1 + Fraction) \times 2^{(Exponent-Bias)}$

$(-1)^0 \times (1 + .0) \times 2^{(24-127)} = 1.0 \times 2^{-103}$

Expanding 1.0×2^{-103} to binary is:

$1.0 \times 2^{-103} = 9.86076132e^{-32}$

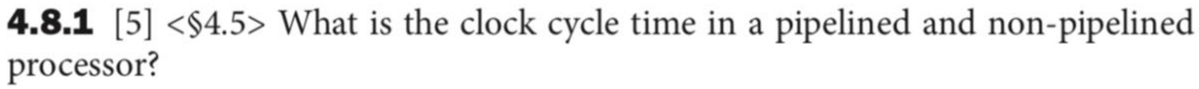


4.8 In this exercise, we examine how pipelining affects the clock cycle time of the processor. Problems in this exercise assume that individual stages of the datapath have the following latencies:

IF	ID	EX	MEM	WB
250ps	350ps	150ps	300ps	200ps

Also, assume that instructions executed by the processor are broken down as follows:

alu	beq	lw	sw
45%	20%	20%	15%

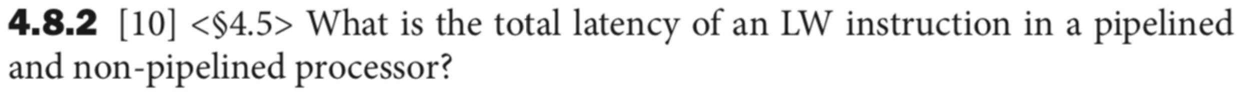


4.8.1 [5] <§4.5> What is the clock cycle time in a pipelined and non-pipelined processor?

In a pipelined processor, a single clock cycle goes through all five stages in which the slowest operation is used which is the ID stage with $\mathbf{350ps.}$

In a non-pipelined processor, every instruction will go through all five stages since there is no pipelining which then we can add to get:

$$\begin{aligned} clock\ cycle\ time &= 250ps + 350ps + 150ps + 300ps + 200ps \\ &= \mathbf{1250ps} \end{aligned}$$



4.8.2 [10] <§4.5> What is the total latency of an LW instruction in a pipelined and non-pipelined processor?

The LW instruction will be used by all five stages.

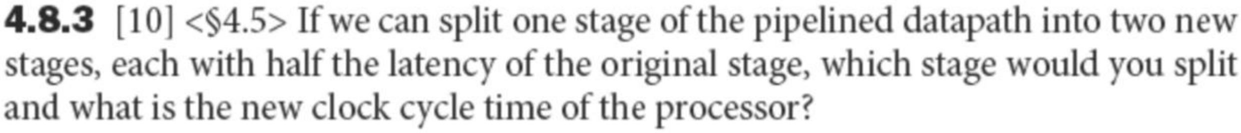
In a pipelined processor, 5 cycles will be processed by the slowest operation which is $350ps$ which the total latency can be found by doing:

$$Total\ Latency = Clock\ Cycle\ Time\ \times\ Cycles$$

$$Total\ Latency = 350ps \times 5 = \mathbf{1750ps}$$

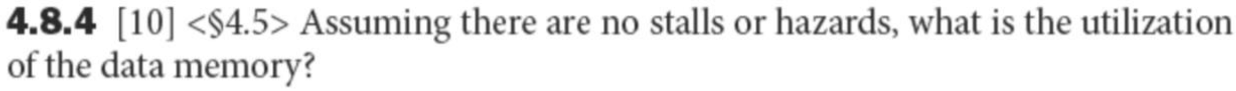
In a non-pipelined processor, the total latency is the sum of all five stages which is:

$$Total\ Latency = 250ps + 350ps + 150ps + 300ps + 200ps = \mathbf{1250ps}$$



4.8.3 [10] <§4.5> If we can split one stage of the pipelined datapath into two new stages, each with half the latency of the original stage, which stage would you split and what is the new clock cycle time of the processor?

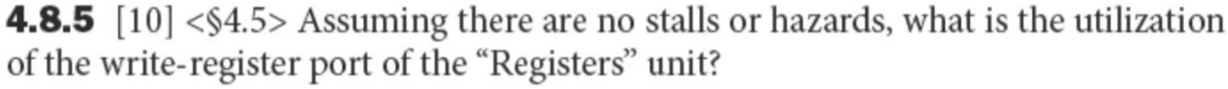
The longest stage has to be split into two new stages in order to decrease cycle time. So, the Instruction Decode (ID) stage is the longest stage with $350ps$, and now the Memory (MEM) stage is the new longest clock cycle time with $300ps$.



4.8.4 [10] <§4.5> Assuming there are no stalls or hazards, what is the utilization of the data memory?

Since LW and SW utilizes the data memory, then the utilization of the data memory is:

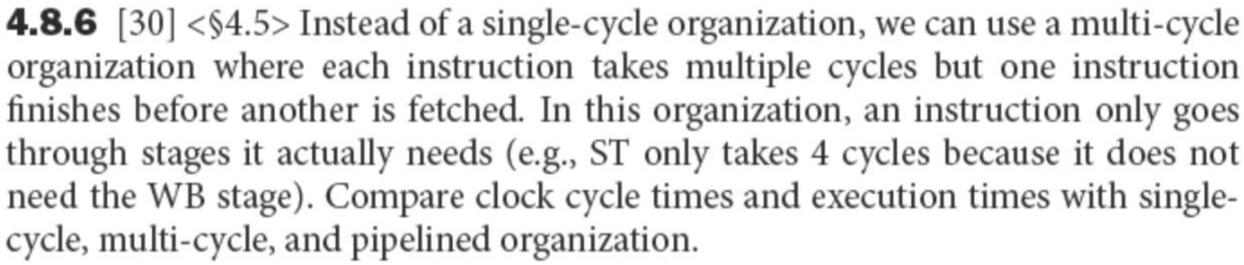
$$20\% + 15\% = \mathbf{35}\%$$



4.8.5 [10] <§4.5> Assuming there are no stalls or hazards, what is the utilization of the write-register port of the "Registers" unit?

Since ALU and LW uses the write-register port of the "Registers" unit, then the utilization of that port is:

$$45\% + 20\% = \mathbf{65}\%$$



4.8.6 [30] <§4.5> Instead of a single-cycle organization, we can use a multi-cycle organization where each instruction takes multiple cycles but one instruction finishes before another is fetched. In this organization, an instruction only goes through stages it actually needs (e.g., ST only takes 4 cycles because it does not need the WB stage). Compare clock cycle times and execution times with single-cycle, multi-cycle, and pipelined organization.

Single-Cycle organization:

$$X = \frac{Clock\ cycle_{non-pipeline}}{Clock\ cycle_{pipeline}} = \frac{1250}{350} = \mathbf{3.5}$$

Where the execution time is the number of X times that goes through the pipeline.

A multi-cycle organization has the same clock cycle time as a pipelined organization. So, in a multi-cycle organization, *LW* completes in 5 cycles, *SW* completes in 4 cycles (no *WB*), an *ALU* instruction completes in 4 cycles (no *MEM*), and *beq* completes in 4 cycles (no *WB*). So, we get the execution time which is the number of X times that goes through the pipeline where X is:

$$X = (5\ \times 20\%) + \left(4 \times (45\% + 20\% + 15\%)\right)$$
$$X = (5\ \times .20) + (4\ \times\ .80)$$
$$X = \mathbf{4.2}$$