Yuqi Mao
page: 1
problem: 1.

1.

instruction	clockcycle	Frequency
All ALU operations	1.0	(39+56)/2 = 47.5%
Loads	3.5	(25+19)/2 = 22%
Stores	2.8	(14+7)/2 = 10.5%
Branches	(4.0+2.0)/2=3.0	(15+15)/2 = 15%
Jumps	2.4	(7+3)/2 = 5%

average CPI

$47.5\% \times 1.0 + 22\% \times 3.5 + 10.5\% \times 2.8 + 15\% \times 3.0 + 5\% \times 2.4$

$= 0.475 + 0.77 + 0.294 + 0.45 + 0.12$

$= 2.109$

Yuqi Mao
page : 2
problem: 2. 3

(1) To calculate the speed from fast mode, its run time without enhancement should be worked. Designers are aware of which 2 selves are implicated throughout the accelerated project planning: 50% unaccelerated part and 50% accelerated part.

Without the enhancement, the unaccelerated part would have taken just as long (50%), but the accelerated part would take 10 times as long, 500%. So the relative execution time without the enhancement would be $50\% + 500\% = 550\%$.

Overall speedup : $$\frac{550\%}{100\%} = 5.5$$

(2). Use these figures in Amdahl's Law

$$\frac{\text{speedup overall} \times \text{speedup accelerated} - \text{speedup accelerated}}{\text{speedup overall} \times \text{speedup accelerated} - \text{speedup overall}}$$

Vectorized fraction $$\frac{5.5 \times 10 - 10}{5.5 \times 10 - 5.5} = \frac{45}{49.5} = 0.9090 = 90.9\%$$

~~speedup overall x speedup~~

3. (i) speedup $= \frac{1}{(1-0.4) + \frac{0.4}{2}} = 1.25$

(ii) speedup $= \frac{1}{(1-0.99) + \frac{0.99}{2}} = 1.98$

(iii) speedup $= \frac{1}{0.2 + 0.8 \times (0.6 + \frac{0.4}{2})} = 1.19$

4 A

(ii) Pseudo-parallelism allows instructions are issued sequentially but the overall execution overlaps, and throughput is higher. Single instruction finishes in a short clock cycle

(iii)

	1	2	3	4	5	6	7	8	9	10	11
	IF	ID	EX	MEM	WB						
		IF	ID	EX	MEM	WB					
			IF	ID	EX	MEM	WB				
				IF	ID	EX	MEM	WB			
					IF	ID	EX	MEM	WB		

(iii) ① from the lw in MEM/WB register

② have a feed back path from this register to one input to data memory

③ have a control signal select the appropriate input to data memory to support this lw-sw input combination

4 B

	1	2	3	4	5	6	7	8	9	10	11	12
	IF	ID	EX									
		IF	ID									
			IF									
				IF	ID	EX	MEM	WB				
					IF	ID	EX	MEM	WB			
						IF	ID	...	EX	MEM	WB	
							IF	...	ID	EX	MEM	WB

Just the sub instruction before it: the sub produces the only data consumed by the add instruction.

Yuqi Mao
page: 4
problem 5

5.

(i) If block size is larger.

- Con: There will be fewer blocks and hence a higher potential for conflict misses
- Pro: Achieve better performance from spatial locality due to the larger block size.
- Example: If we have a high degree of sequential data accesses, this make more sense

If there are fewer elements per block and more blocks:

- Con: We may be more subject to compulsory misses due to the smaller block size.
- Pro We may see fewer conflict misses due to more unique mappings.
- Example: If we have more random memory accesses, this makes more sense.

(ii) Given that the physical address is 20 bits long. and the tag is 11 bits, there are 9 bits left over for the index and offset.

We can determine the number of bits of offset as the problem states that:

① Data is word addressable and words are 8 bits long
② Each block holds 16 bytes

As there are 8 bits PI, each block holds 16 words, thus 4 bits of offset are needed. There are 5 bits left for the index. Thus, there are 2^5 (32) blocks in the cache.

Yuqi Mao

5.(iii)

Yuqi Mao
page : 5
problem : 5.

We can calculate the number of bits for the offset first:

① There are 16 data words per block which implies that at least 4 bits are needed.

② Because data is addressable to the $\frac{1}{2}$ word, an additional bit bit of offset is needed

③ Thus the offset is 5 bits

To calculate the index, we need to use the information given regarding the total capacity of the cache

① 2MB $= 2^{21}$ total bytes

② Use this information to determine the total number of blocks in the cache.

$2^{21} \times (1/16) \times (1/64) \times (8/1) = 2^{14}$ blocks.

③ There are 2^4 (16) blocks/set.

$2^{14} \times (1/2^4) = 2^{10}$ sets.

④ Thus 10 bits for the index.

The remainig bits form the tag

$64 - 5 - 10 = 49$ bits 49 bits for tag

In conclusion Tag: 49 bits ; Index: 10bits . Offset: 5bits

IF ID ... EX MEM WB