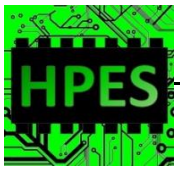

EEE4120F Quiz 2 based on paper:

"Amdahl Law in Multicore Era" by Hill and Marty **DATE:** 5/3/2020

Name: _____ Student Number: _____ *Fill in name!*

*This is just a very short quiz, but it **is** for marks!*

NB: Please select only one answer option for each question

CIRCLE/COLOR-IN ANSWERS FOR MULTIPLE CHIOCE QUESTIONS **TIME (mins): 20**

TOTAL NUMBER OF QUESTIONS : FIVE (5)

#	Question - EACH QUESTION WORTH 1 MARK	Mark	Mins	
Q1		1	1	X
	Hill and Marty discuss Amdahl's classic law, which is typically taught in a university curriculum, expresses speedup as ... (choose the right option:) [1] Let speedup be the difference between the original execution time and the enhanced execution time. **[2]** Let speedup be the original execution time divided by the enhanced execution time. **<==** [3] Let speedup be the enhanced execution time divided by the original execution time. [4] Let speedup be 1 over the original execution time minus the enhanced execution time. [5] Let speedup be 1 over the sum of: the original execution time divided by the enhanced execution plus the enhanced execution time. (*terminology note:* consider the "original execution time" to be the same as the "sequential" or "non-enhanced execution time.)			(marking column)
Q2	Here's an easy gift for you…. Explain the term BCE in relation to this paper:	1	0.5	X
	[1] BCE = Base Core Equivalent **<==** [2] BCE = Basic Core Equivalent [3] BCE = Basic Component Entity [4] BCE = Baseline Core Element [5] BCE = Base Compute and Element			
Info for Q3 - Q5	The authors use the term "more powerful core" or "larger core" to refer to an enhanced core that may take multiple BCEs in terms of resources; I like to use the term "megacore" because that's more commonly used at least in relation to Xilinx IP. So a BCE and a megacore doesn't necessarily mean they have to be CPUs that run a standard set of instructions. That leads is in the main question for this test.... Please see the attached handout with the 1-BCE vs 4-BCE processor descriptions.... and answer these questions			
Q3	The summing procedure shown on the handout is the loop we want to run most of the time, i.e. the 7 instructions shown for the Loop in Table 1. Both implementations of Loop for the 1-BCE and the 4-BCE are equivalent comprising 7 instructions. Let's compare a multicore chip that has 4x 1-BCEs running Loop in parallel (with F set to a different start value for each thread) to a chip with just a single 1x 4-BCE. What is the speedup of the 4x 1-BCE running Loop in parallel over the 1x 4-BCE running one instance of Loop?	2	2	X
	[1] 1 [2] 1/2 [3] 1/4 **[4] 2 <==** [5] 4			

Please turn over ….

Q4	Asystemmatic speedup is defined as: $$\text{Asymmetric Speedup} = \frac{1}{\frac{1-F}{\text{perf}(R)} + \frac{F}{\text{perf}(R) + (n-R)}}$$ R = 4, considering that a megacore consume 4x BCEs If we consider the parallel portion is 80% and a chip that contains 6 BCE resources, what would be the asymmetric speedup were we to have 1x 4-BCE and 2x 1-BCEs all running the Loop in parallel?		2	2	X
	[1] between 1 and 2 [2] between 2 and 3. **[3]** between 3 and 4. **<==** [4] between 4 and 5 [5] more than 5				
Q5	This is an opportunity to demonstrate your assembly prowess and another speedup calculation (the question is not worth so much but gives a chance to reach the stratospheric level of marks). Contemplate the MISC2 instruction set and the program in Table 1 of the handout. Propose a better optimized assembly implementation using a better choice of MISC2 instructions. Write your solution in the space below, and indicate what speedup of the Loop your implementation running on 1x 4-BCE would give in comparison to the original Loop running on 1x 1-BCE. (if using additional page, please put student number at the top of page!)		4	4	X

	Suggested implementation:
	
	`CLR B ; clear the sum register`
	`SET A,#1000 ; set A to memory starting point`
	`Loop: LD C,[A+=4] ; C = mem[A]; A = A + 4;`
	`LD D,[A+=4] ; D = mem[A]; A = A + 4;`
	`ADD B,B,C ; B = B + C;`
	`ADD B,B,D ; B = B + D;`
	`J Loop ; repeat`
	
	So the loop is taking 4 instructions, compared to 7 instructions that the MISC1 core required. That is a speed up of 7/5 in terms of instructions. It is also doing 2 words in one loop so that is (7x2)/5. But also rememeber that the 4-BCE is 2x as fast as the 1-BCE so the speedup is then going to be:
	(7x2x2)/5 = **28/5** = **5.6**
	So in other words by adjusting the programme to accommodate the more enhanced instructions, the 4-BCE achieves a speedup of 5.6!				

			TOTAL :	10	

Time : time est. in minutes, Mark : marks that question is worth X : for office use