EEE4084F Quiz 1: Lectures 1-3

24-Fel

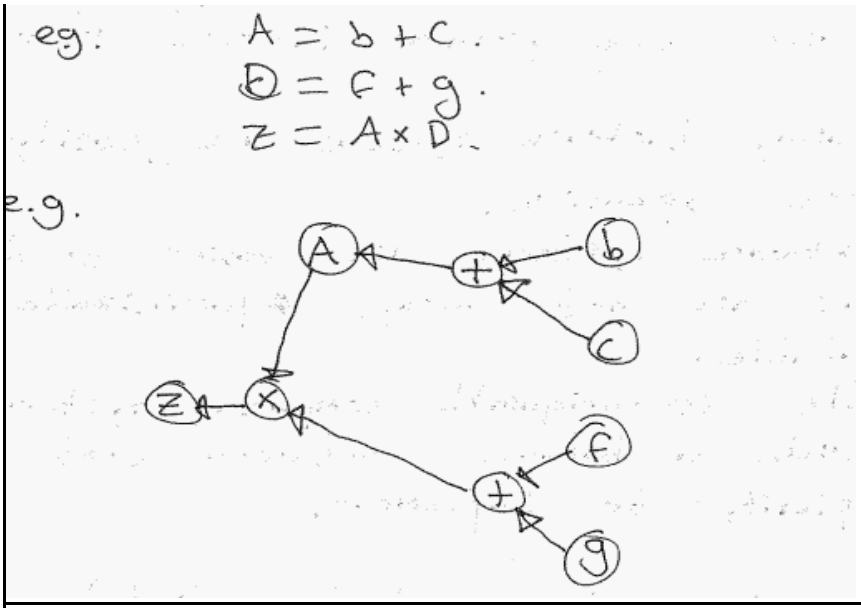
!!! SOLUTION !!!

TOTAL NUMBER OF QUESTIONS : FIVE

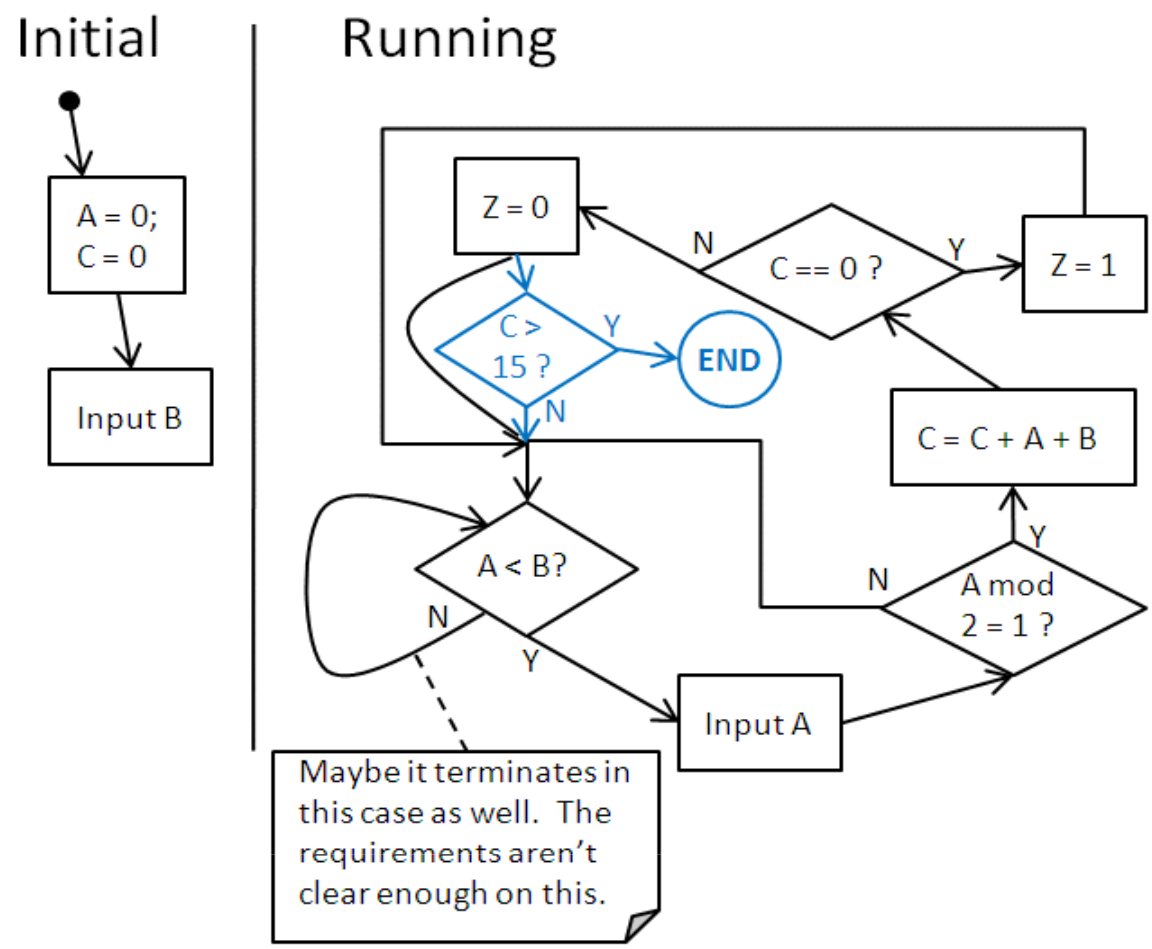
time(min)

#	Question	Marks
Q1	(a) Give two examples of mainstream applications that are meaningful to implement using a parallel computing architecture. [2 marks] (b) For each application mentioned for part (a) motivate why this would be suited to a parallel architecture. [2 x 2 = 4 marks]	6
	(a) Weather prediction, web servers, social networking, scientific modelling, database queries, code breaking, image processing (e.g., blur filters, edge detection), amount other applications. (b) In general the answer should highlight a property of the application in which the can be a large amount of data to process, and for which the processing can be done concurrently. For example in code breaking, as many as possible (# of processing units) can simultaneously try different keys to decrypt a message. For web serving, it may be handling many requests at the same time by different processors in order to allow a higher overall throughput and lower	
Q2	Discuss benefits and drawbacks of the different types of computing approaches, i.e. contrasting between 1) a purely hardware implementation, 2) a solution that involves software on a fixed architecture, and 3) a reconfigurable computing platform. (for this question you do not have to list a total of 6 points, some of the marks are allocated to structuring meaningful and logical sentences). [6 marks]	6
	Below benefits shown as + and drawbacks shown at - Purely hardware: - static / difficult to change (which out major manual intervention) + fast, not necessarily reliant on a particular clock speed. + allows a highly parallel solution - can be more expensive for (small runs) and at the prototyping stage (one mistake could mean a total refabrication of PCBs) Microprocessor-based / software solution: + more flexibility than hardware along + can turn out cheaper (esp. for prototyping or small number of products) as in COTS / PC computer platform - software coding can add expense and time that doesn't afflict a purely hardware case. + faster turn-around time for debug and test cycle - limitations of sequential processing (for single-processor platform) - could be entirely overpowered and power uneconomical than the case of a hardware only solution Reconfigurable computing + Provides benefits of both 'worlds' (i.e., harware and software solutions), such as faster turn-around time (in terms of the FPGA-based hardware) and the software parts. Can be cheapter in terms of hardware if using COTS reconfigurable comp platform. + More flexible than the other solutions (can change both h/w & s/w without redoing PCB) - Can be more expensive both in terms of hardware (if building a specialized platform) and in dealing with software / HDL code. - Can become significantly more complex and this time-consuming and costly having to deal	

Q3	(a) What is meant by the term 'automatic parallelism'? [2 marks] (b) Discuss how such a process might be achieved. [4 marks] (c) Mention three drawbacks or challenges that would need to be overcome in order to make automatic parallelism effective. [3 marks]	9
	(a) Automatic parallelism refers to the process by which a sequential program is automatically converted into a prallel one that can be run on a multiprocessor platform. (b) An automatic parallelism translation program would need to interpret large pieces of a program in order to obtain an 'understanding' of what the code is attempting to achieve (including finding data dependencies and areas of independence) in order to re-represent it in a parallel solution. In comparison, a traditional compiler usually processes code one statement at a time, developing a narrower impression of what the program is attempting to achieve. (c) A main drawback to automatic parallelism is the unpredictability of the resulting parallel solution; for example, a particular sequential code may be represented in a number of different ways by different translators (or different versions of the same translator). Also portability may be a problem; for example one sequential program might not work (or work inefectively) on different architectures. There may be debugging difficulties; for example, how would a breakpoint in the sequential code relate to the parallel version; or how would single-stepping be done; or how would watch variables work? The translator would also need to effectively contend with the various parallel programming challenges, such as synchronization, deadlock, inter-process communications, critical sections, etc.	
Q4	(a) Briefly explain the difference between the spatial and the temporal computing paradigms. (Optional rough sketch to aid your explanation.) [4 marks] (b) Briefly discuss why you think it may would be nice if parallel programs could be effectively coded using the temporal paradigm. [3 marks]	7
	(a) Temporal computing is an approach by which a program is represented by a sequence of ordered steps, where steps are carried out in chronological order (i.e., "things done over a time period"). Spatial computing concerns relations, dependencies and transfers between computing elements; in this approach computation is expressed in spaces (more colloqually, "things related in space"). (b) This question doesn't have a unique correct solution. A likely reason is that: Most people nowadays tend to be taught programming (or more generally, the way to describe a solution) in the traditional sense of using a series of steps, usually one step following the next -- this is likely based on the first-person perspective where a person can basically do just one task at a time. Thus, for people educated in this way, it comes more naturally to write a program in this way as well; for the same reason, the code is easier for others to understand as well. Thus, if it's possible to use this easier approach, developers can complete tasks more quickly and be more easily able to understand what they have done in past projects, or what others have done in other projects	
	The scanned snippet below, from a student's answer, provides a pretty good example of answering question (a) with the inclusion of a clear example.	



Q5	Consider the following application requirements. Note that these requirements are not in a sequential order, you need to consider the dependencies and relations between them. - Create an internal integer variable C that is initially set to 0. A = 0 initially. - The program must input two integer values, A, B - The program must output a value Z that is either 0 or 1 - Input a value for B at the start of the program - While A is less than B, input a new A - When a new A has been input and A is an odd number, set variable C to C + A + B - Whenever C changes, and C is zero, set output Z to 1; set output Z to 0 otherwise. - If C is greater than 15 end the program *<== this line was missing in the printout!*	17
	Complete the following for Q5:	
	(a) Provide a spatial representation of the program described above. Attempt to describe your solution as structurally clearly as you can. [10 marks]	
	(b) Discuss potential drawbacks, and possible improvements, to the way the requirements for this program are documented. Consider the scalability of this approach, as it how it would influence a larger and more complex application. [4 marks]	
	(c) Consider the following scenario and decide if the program terminates. If the program terminates, what will the final value of Z be (i.e. 1 or 0)? Assume 32bit integers are used. [3 marks]	
	input B is set to 5	
	input A is set to 3	
	input A is set to 1	
	input A is set to 2	
	input A is set to 2	
	input A is set to 8	
	Comment: Unfortunately the last line of the requirements list didn't come out in the printout, which made the whole question more confusing.But the solution will work on the case that the last line was missing.	
	(a) The diagram below provides a spatial representation of a program based on the requirements given. The part in blue indicates what would be added if the last line was in the requirements. I chose to describe this as a flowchart (although these are generally used to represent steps of a program, it is also a kind of spatial representation).	



(b) A potential drawback is that it is largely written in a natural language which can make the description lengthy and imprecise for the purpose of accuratly and concisely describing the requirements of such a system. Furthermore, there are aspects of the program that are left unclear. For example, what should be done if B is 0 at the start? Should the program continue to spin without anything happening although there's no chance that an A will be input? Improvement would be to use a more suitable language to describe the requirements, possibly a kind of pseudo code or just C since the description given is fairly close to the level a standard program language is at. Another improvement would be clarification of the termination condition and illimination of ambiguities.

(c) The program is likely not to terminate because there is no terminating condition; according to the input given, the program will sit waiting at the Input A state and will continue waiting for another input to be given. If the C>15 termination condition was included, it wouldn't have been reached because the program would also be waiting for an A input in that version of the requirements. (If, however, the requirements indicated that the program ends after the last input, then the program would terminate with C =14 and Z = 0.)

TOTAL : 45

Time : time est. in sec W : Weighting of question % : How much question counts X : Office use