Brown University
School of Engineering
EN1640 Design of Computing Systems
Professor Sherief Reda
HW 04. 25 points. Due Date: Wednesday April 20th in class.

1. (15 points) We consider the execution of a loop in a *single-cycle* VLIW processor. Assume that any combination of instruction types can be executed in the same cycle. Note that such assumption only removes resource constraints, but data and control dependencies must be still handled correctly. Assume R0 has the value 0. Assume that register reads occur before register writes in the same cycle.

```
loop: LDR   R1, [R6, 40]
      ADD   R5, R5, R1
      STR   R1, [R5, 20]
      ADD   R6, R6, 4
      add   R5, R5, -4
      CMP   R5, R0
      BE    LOOP
```

a. [10 points] Unroll this loop once and schedule it for a 2-issue static VLIW processor. Assume that the loop always executes an even number of iterations. Hint: for correct scheduling, you need to identify all potential data and control hazards. Use register renaming whenever possible to eliminate false dependencies.
b. [5 points] What is the speed-up of using your scheduled code instead of the original code? Assume that the loop will be executed a very large number of times.

2. Consider a simple 4-stage pipelined processor design that is fully-bypassed so it never has to stall for data dependence. The four stages are IF (instruction fetch), DE (decode), EX (execute), WB (write-back). The critical path delays in the 4 stages are 5ns, 4ns, 10ns, and 3ns, respectively. A proposal to improve the performance of the processor is to further pipeline the EX stage into two 5ns stages, EX1 and EX2. Source operands of an instruction are required at the start of EX1; the result of an instruction is only available for forwarding at the end of EX2.

a. (3 points) Consider an instruction stream segment consisting of …; I_A; I_B; I_C; I_D; I_E; …where I_B and I_C have back-to-back data dependence. All other instruction pairs are dependence free. Complete the following Pipeline Resource Diagram for I_A, I_B, I_C, I_D, and I_E.

	t_0	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}	t_{11}	t_{12}	t_{13}
IF	I_A	I_B												
DE		I_A												
EX1			I_A											
EX2				I_A										
WB					I_A									

b. (3 points) The stall caused by back-to-back data-dependency between instructions cannot be eliminated by further bypassing. But IPC can be restored to 1 if two independent threads are interleaved (fine-grain) in the pipeline. Consider two segments …, $I_{1:A}$; $I_{1:B}$; $I_{1:C}$,… and ….,$I_{2:A}$; $I_{2:B}$; $I_{2:C}$,… from two independent instruction streams. Instructions $I_{1:B}$ and $I_{1:C}$ have back-to-back data dependence. Instructions $I_{2:A}$ and $I_{2:B}$ also have back-to-back data dependence. All other instruction pairs are dependence free. Complete the following Pipeline Resource Diagram for their execution on an interleaved pipeline.

	t_0	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}	t_{11}	t_{12}	t_{13}
IF	$I_{1:A}$													
DE		$I_{1:A}$												
EX1			$I_{1:A}$											
EX2				$I_{1:A}$										
WB					$I_{1:A}$									

c. (2 points) Values written to the register file in the WB stage is only available for reading by the DE stage in the next cycle. If we wish to eliminate all forwarding paths from the processor, how many threads must be interleaved in the new datapath to maintain IPC=1?