Homework-4

1. Add NOP instructions to the code below so that it will run correctly on a pipeline that does not handle data hazards.

```
ADDI X1, X2, #5
ADD X3, X1, X2
ADDI X4, X1, #15
ADD X5, X3, X2
```

Homework-4

2. Problems in this exercise assume that the logic blocks used to implement a processor's datapath have the following latencies:

I-Mem / D-Mem	Register File	Mux	ALU	Adder	Single gate	Register Read	Register Setup	Sign extend	Control
250 ps	150 ps	25 ps	200 ps	150 ps	5 ps	30 ps	20 ps	50 ps	50 ps

"Register read" is the time needed after the rising clock edge for the new register value to appear on the output. This value applies to the PC only. "Register setup" is the amount of time a register's data input must be stable before the rising edge of the clock. This value applies to both the PC and Register File.

(1) Although the control unit as a whole requires 50ps, it so happens that we can extract the correct value of the Reg2Loc control wire directly from the instruction. Thus, the value of this control wire is available at the same time as the instruction. Explain how we can extract this value directly from the instruction. Hints: Carefully examine the opcodes shown in Figure 2.20. Also, remember that LSR and LSL do not use the Rm field. Finally, ignore STXR.

MK MORGAN KAUFMANN

Homework-4

(2)What is the latency of an R-type instruction (i.e., how long must the clock period be to ensure that this instruction works correctly)?
(3)What is the latency of LDUR? (Check your answer carefully. Many students place extra muxes on the critical path.)
(4)What is the latency of STUR? (Check your answer carefully. Many students place extra muxes on the critical path.)
(5)What is the latency of CBZ?
(6)What is the latency of B?
(7)What is the latency of an I-type instruction?
(8)What is the minimum clock period for this CPU?

Homework-4

Instruction	Opcode	Opcode Size	11-bit opcode range		Instruction Format
			Start	End	
B	000101	6	160	191	B - format
STURB	00111000000	11	448		D - format
LDURB	00111000010	11	450		D - format
B.cond	01010100	8	672	679	CB - format
ORRI	1011001000	10	712	713	I - format
EORI	1101001000	10	840	841	I - format
STURH	01111000000	11	960		D - format
LDURH	01111000010	11	962		D - format
AND	10001010000	11	1104		R - format
ADD	10001011000	11	1112		R - format
ADDI	1001000100	10	1160	1161	I - format
ANDI	1001001000	10	1168	1169	I - format
BL	100101	6	1184	1215	B - format
ORR	10101010000	11	1360		R - format
ADDS	10101011000	11	1368		R - format
ADDIS	1011000100	10	1416	1417	I - format
CBZ	10110100	8	1440	1447	CB - format
CBNZ	10110101	8	1448	1455	CB - format
STURW	10111000000	11	1472		D - format
LDURSW	10111000100	11	1476		D - format
STXR	11001000000	11	1600		D - format
LDXR	11001000010	11	1602		D - format
EOR	11101010000	11	1616		R - format
SUB	11001011000	11	1624		R - format
SUBI	1101000100	10	1672	1673	I - format
MOVZ	110100101	9	1684	1687	IM - format
LSR	11010011010	11	1690		R - format
LSL	11010011011	11	1691		R - format
BR	11010110000	11	1712		R - format
ANDS	11101010000	11	1872		R - format
SUBS	11101011000	11	1880		R - format
SUBIS	1111000100	10	1928	1929	I - format
ANDIS	1111001000	10	1936	1937	I - format
MOVK	111100101	9	1940	1943	IM - format
STUR	11111000000	11	1984		D - format
LDUR	11111000010	11	1986		D - format

FIGURE 2.20 LEGv8 instruction encoding. The varying size opcode values can be mapped into the space they occupy in the widest opcodes. By looking at the first 11 bits of the instruction and looking up the value, you can see which instruction it refers to.

MK MORGAN KAUFMANN