Book Questions

1.16) 6 points Total: 2 points each

1) The set of instructions the CPU can carry out (operands)
2) Acceptable representations of operands (data types)
3) The mechanisms that the computer can use to figure out where the operands are located. (addressing modes)

1.17) 5 points

An ISA is a specification for what the underlying hardware needs to be able to carry out. A microarchitecture is the hardware implementation of a given ISA.

(Extra Info to help understanding – not necessary for full credit)
So, there can be many different microarchitectures for a given ISA, but there can only be one ISA for a given microarchitecture.

1.23) 5 points

It is unlikely to change because older software would not work if it did. The executables use the set of operands for a given ISA. If the ISA changes, these executables can no longer run. So, ISAs are unlikely to change because nobody wants to be forced to buy new/create new software each time a new microarchitecture comes out.

4.6) 4 points

Opcode: What the instruction does (ADD, AND, BRANCH, etc.)
Operands: What the operations are performed on

Custom Questions

2abc) 50 Points Total: (464 cells X 0.1 point per cell + 3.6 points free for doing it)

see attached spreadsheet.

2) 30 points Total: Answers vary.
10 points for Comments, 10 points for instructions, 10 points for binary

```
while (B > 0)
{
        C=C+A
        B=B-1
}
```

R2 = A
R3 = B
R4 = C

NB: Colors are used to show which bit correspond to which part of the instruction. The colors do **NOT** indicate whether an operand is a D, S, or T register.

Assumes R2 and R3 have been loaded with the values to multiply

Address	Program Memory Contents	Instructions	Comment
0	1001100000000000	CONST R4, x00	C=0
1	1101100100000000	HICONST R4, x00	C=0
2	0010011110000000	CMPIU R3, #0	put B in NZP
3	0000110000000011	BRnz 3	if b=<0, go to instruction 7
4	0001100100000010	ADD R4, R4, R2	C=C+A
5	0001011011111111	ADD, R3, R3, -1	B=B-1, and puts B in NZP
6	0000001111111101	BRp -3	if b>0, go to instruction 4
7	0000111111111111	BRnzp -1	infinite loop

Extra Credit)
2.5 Points (All or Nothing)

Register	Contents
R1	0x4022
R2	0xEF24
R3	0xEF25
R4	0xEF26

Control Settings

	PCMux.CTL	rsMux.CTL	rtMux.CTL	rdMux.CTL	regFile.WE	regInputMux.CTL	Arith.CTL	ArithMux.CTL	LOGIC.CTL	LogicMux.CTL	SHIFT.CTL	COSNT.CTL	CMP.CTL	ALUMux.CTL	NZP.WE	DATA.WE
BRxxx	0	x	x	x	0	x	x	x	x	X	x	x	x	x	0	0
ADD	1	0	0	0	1	0	0	0	x	X	x	x	x	0	1	0
MUL	1	0	0	0	1	0	1	0	x	X	x	x	x	0	1	0
SUB	1	0	0	0	1	0	2	0	x	X	x	x	x	0	1	0
DIV	1	0	0	0	1	0	3	0	x	X	x	x	x	0	1	0
ADD (immediate)	1	0	x	0	1	0	0	1	x	X	x	x	x	0	1	0
CMP	1	2	0	x	0	0	x	x	x	X	x	x	0	4	1	0
CMPU	1	2	0	x	0	0	x	x	x	X	x	x	1	4	1	0
CMPI	1	2	0	x	0	0	x	x	x	X	x	x	2	4	1	0
CMPIU	1	2	0	x	0	0	x	x	x	X	x	x	3	4	1	0
JSR	5	x	x	1	1	2	x	x	x	X	x	x	x	x	1	0
JSRR	3	0	x	1	1	2	x	x	x	X	x	x	x	x	1	0
AND	1	0	0	0	1	0	x	x	0	0	x	x	x	1	1	0
NOT	1	0	x	0	1	0	x	x	1	X	x	x	x	1	1	0
OR	1	0	0	0	1	0	x	x	2	0	x	x	x	1	1	0
XOR	1	0	0	0	1	0	x	x	3	0	x	x	x	1	1	0
AND (immediate)	1	0	0	0	1	0	x	x	0	1	x	x	x	1	1	0
LDR	1	0	x	0	1	1	0	2	x	x	x	x	x	0	1	0
STR	1	0	1	x	0	x	0	2	x	x	x	x	x	0	0	1
RTI	3	1	x	x	0	x	x	x	x	x	x	x	x	x	0	0
CONST	1	x	x	0	1	0	x	x	x	x	x	0	x	3	1	0
SLL	1	0	x	0	1	0	x	x	x	x	0	x	x	2	1	0

SRA	1	0	x	0	1	0	x	x	x	x	1	x	x	2	1	0
SRL	1	0	x	0	1	0	x	x	x	x	2	x	x	2	1	0
MOD	1	0	0	0	1	0	4	0	x	x	x	x	x	0	1	0
JMPR	3	0	x	x	0	x	x	x	x	x	x	x	x	x	0	0
JMP	2	x	x	x	0	x	x	x	x	x	x	x	x	x	0	0
HICONST	1	2	x	0	1	0	x	x	x	x	x	1	x	3	1	0
TRAP	4	x	x	1	1	2	x	x	x	x	x	x	x	x	1	0