CO450 Computer Architectures Week 6 Exercise Handout

Recap on Signed Magnitude Notation

1. Represent the following decimal number in binary using Signed Magnitude Notation:

-97_{10}

128	64	32	16	8	4	2	1
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	1	1	·	·	·	·	1

The correct answer is:

1	1	1	·	·	·	·	1

Recap on Binary Excess Notation to Decimal

1. What is the decimal number that is represented by 10101110_2 in Excess Notation?

	128	64	32	16	8	4	2	1
	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
	1	·	1	·	1	1	1	·
Unsigned binary to decimal conversion using positional notation	174 -128							
Unsigned decimal value minus Excess ($2^{(n-1)}$) Note: *n* = number of bits	46							

The correct answer is:

46

Recap on Decimal to Binary Excess Notation

1. What is the binary Excess Notation representation of the following decimal number:

-15_{10}

Decimal plus Excess ($2^{(n-1)}$) Note: *n* = number of bits	-15 +128 = 113							
	128	64	32	16	8	4	2	1
	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Convert Decimal with Excess to binary using positional notation	·	1	1	1	·	·	·	1

The correct answer is:

·	1	1	1	·	·	·	1

Recap on Two's Complement

1. Convert 74_{10} to binary then use Two's Complement to convert the unsigned binary representation of 74_{10} in to the Two's Complemented binary representation for -74_{10}, what is the correct answer:

	128	64	32	16	8	4	2	1
	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Positional notation used to convert decimal to binary	·	1	·	·	1	·	1	·
Flipped bits	1	·	1	1	·	1	·	1
One to add to the flipped bits above								1
Result of addition of flipped bits and one	1	·	1	1	·	1	1	·
Carry Bits							1	

The correct answer is:

1	·	1	1	·	1	1	·

Recap on Two's Complement Binary Additions

1. Add the following numbers together using two's complement binary representation and then answer the questions below:

67_{10} + -36_{10} =

			128	64	32	16	8	4	2	1
			2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
			·	1	·	·	·	·	1	1
+			1	1	·	1	1	1	·	·
			·	·	·	1	1	1	1	1
		1	1							

Did the calculation produce an overflow? YES / NO

Did the calculation produce a carryout? YES / NO

Would the calculation produce a correct result in an 8 bit system? YES / NO

How many bits were carried to the left during the calculation? 2

Recap on Decimal to Excess 50 Notation

1. Represent -34_{10} in excess 50 notation:

We have done this first one for you.

-34	+	50_{10}	=	16

The correct answer is:

16

Recap on Conversion of Decimal Numbers to SEEZMMMM Format

1. Convert -435.679$_{10}$ into the SEEZMMMM format:

We have done this first one for you.

S = signed magnitude notation	=	1
EE = Exponent Excess 50 notation	=	53
ZMMMM = Mantissa	=	.435679
Normalise Mantissa	=	43568

The correct answer is:

1 53 43568

Recap on Conversion of SEEZMMMM Format to Decimal Number

1. Convert 15322856 in the SEEZMMMM Format to a decimal number:

We have done this first one for you.

S = signed magnitude notation	=	1	=	−
EE = Exponent Excess 50 notation	=	53	=	3
ZMMMM = Mantissa	=	22856	=	

The correct answer is:

−228.56

Recap on Conversion of Decimal Exponent to Excess 127 Binary

1. Represent an Exponent of -6$_{10}$ in the Excess 127 format:

Excess	127	=	.	1	1	1	1	1	1	1
Two's Complement Exponent	-6	=	1	1	1	1	1	.	1	.
Result of addition of excess and exponent		1	.	1	1	1	1	.	.	1
Carry's		1	1	1	1	1	1	1		

The correct answer is:

.1111..1	=	121

127−6

So box was pointless

Recap on Conversion of IEEE 754 Single Precision Binary Float to Decimal

1. Convert the following IEEE 754 single precision binary float to decimal:

$11000010110110110000000000000000_2$

We have done this one for you.

Sign		=	−
Excess		=	1 . . . . 1 . 1
Convert excess binary unsigned to decimal and subtract excess 127 to get exponent		=	133 − 127 = 6

Mantissa	1.1011011
Shift decimal point with exponent	1 1011.1
Convert shifted binary mantissa to decimal	109.5
Add sign to converted decimal	−109.5

The correct answer is:

−109.5

The Little Man Computer: First Program

Accessed at:

Look at the code below and complete the table description to explain what the code is doing at each line?

```
INP
OUT
HLT
```

Line	Code	Description
00	INP	getting a user input
01	OUT	printing the input
02	HLT	stopping running

The Little Man Computer: Second Program

Look at the code below and complete the table description to explain what the code is doing at each line?

```
            INP
            STA   first
            INP
            STA   second
            LDA   first
            OUT
            LDA   second
            OUT
            HLT
first       DAT
second      DAT
```

Line	Code	Description
00	INP	user inputs number
01	STA first	number is stored as first
02	INP	user inputs number
03	STA second	number stored as second
04	LDA first	loads first number
05	OUT	outputs first number
06	LDA second	loads second number
07	OUT	outputs second number
08	HLT	stops program
09	first DAT	creates variable first
10	second DAT	creates variable second

The Little Man Computer: Third Program

Look at the code below and complete the table description to explain what the code is doing at each line?

```
        INP
        STA   first
        INP
        ADD   first
        OUT
        INP
        SUB   first
        OUT
        HLT
first   DAT
```

Line	Code	Description
00	INP	user inputs number
01	STA first	stored as first
02	INP	inouts number
03	ADD first	first added to it
04	OUT	prints answer
05	INP	user inputs number
06	SUB first	first number subtracted from it
07	OUT	prints answer
08	HLT	stops program
09	first DAT	creates variable first

The Little Man Computer: Fourth Program

Look at the code below and complete the table description to explain what the code is doing at each line?

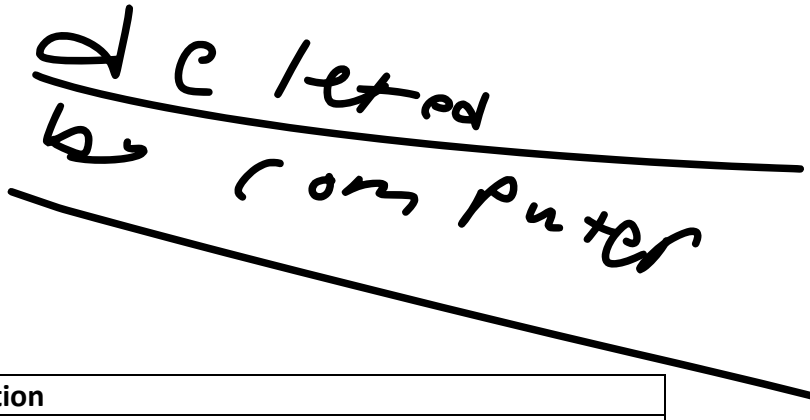
```
            INP
            STA   first
            INP
            STA   second
            SUB   first
            BRP   secondbig
            LDA   first
            OUT
            BRA   stop
secondbig   LDA   second
            OUT
stop        HLT
first       DAT
second      DAT
```

Line	Code	Description
00	INP	user inputs number
01	STA first	its stored as first
02	INP	user inputs number
03	STA second	its stored as second
04	SUB first	second - first
05	BRP secondbig	branch if positive or 0 to secondbig
06	LDA first	loads first number
07	OUT	outputs first number
08	BRA stop	branch to stop
09	secondbig LDA second	defined as secondbig, loads second number
11	OUT	prints second number
11	stop HLT	defined as stop, stops program
12	first DAT	creates variable first
13	second DAT	creates variable second

The Little Man Computer: Fifth Program

Look at the code below and complete the table description to explain what the code is doing at each line?

```
            LDA   one
            STA   count
            OUT
loop        LDA   count
            ADD   one
            OUT
            STA   count
            SUB   ten
            BRP   stop
            BRA   loop
stop        HLT
one         DAT   001
ten         DAT   010
count       DAT
```



Line	Code	Description
00	LDA one	
01	STA count	
02	OUT	
03	loop LDA count	
04	ADD one	
05	OUT	
06	STA count	
07	SUB ten	
08	BRP stop	
09	BRA loop	
10	stop HLT	
11	one DAT 001	
12	ten DAT 010	
13	count DAT	

The Little Man Computer: Programming Problems

1. Using the Little Man Computer design a program to multiply any number entered by a user by 10 and output the result.

Hint: There is no Multiplication command in the Little Man Computer, but you can Loop as many times as you want, so you will need to use Branching within your solution.

2. Using the Little Man Computer design a program to output how many fives will go exactly into any number entered by a user (you do not need to allow for any remainders).

Hint: There is no Divide command in the Little Man Computer, but you can Subtract as many times as you want.

The Little Man Computer: Further Challenging Programming Problems

Now that you have started to understand the Little Man Computer and its Assembly Language you should try developing programming solutions to some more challenging problems.

Develop programs in the Little Man Computer to:

1. Output Fibonacci numbers (a series of numbers in which each number (Fibonacci number) is the sum of the two preceding numbers. The simplest is the series 1, 1, 2, 3, 5, 8, etc.)
2. Output the first *n* Fibonacci numbers, where *n* is a user input
3. Take two number inputs *a, b* and compute *a* x *b*
4. Take a number input *a* and compute *a* divided by 2
5. Take two number inputs *a, b* and compute a divided by *b*
6. Take number user inputs until an input of 0 is received, then output the smallest of the inputs
7. Take two number inputs and output the highest common factor using Euclid's algorithm (an algorithm for finding the greatest common divisor of two numbers *a* and *b*)

The Answers

Signed Magnitude Notation

1. 11100001_2

Binary Excess Notation to Decimal

1. $+46_{10}$

Decimal to Binary Excess Notation

1. 01110001_2

Two's Complement

1. 10110110_2

Two's Complement Binary Additions

1. No, Yes, Yes, 2

Decimal to Excess 50 Notation

1. $+16_{10}$

Conversion of Decimal Numbers to SEEZMMMM Format

1. 15343568

Conversion of SEEZMMMM Format to Decimal Number

1. -228.56_{10}

Conversion of Decimal Exponent to Excess 127 Binary

1. $01111001_2 = 121_{10}$

Conversion of IEEE 754 Single Precision Binary Float to Decimal

1. -45.5_{10}

The Little Man Computer: Programming Problems

1.

```
            INP
            STA   value
loop        LDA   product
            ADD   value
            STA   product
            LDA   loopcount
            SUB   deductor
            STA   loopcount
            BRP   loop
            LDA   product
            OUT
            HLT
value       DAT
loopcount   DAT   009
product     DAT   000
deductor    DAT   001
```

2.

```
            INP
            STA   value
loop        LDA   count
            ADD   increment
            STA   count
            LDA   value
            SUB   divisor
            STA   value
            BRP   loop
            LDA   count
            SUB   increment
            OUT
            HLT
value       DAT
divisor     DAT   005
count       DAT   000
increment   DAT   001
```