ON FINAL !! Whole Page

Question 1 (3 Marks)

Perform the following conversions:

256 128 64 32 16 8 4 2 1
173 1 0 1 0 1 1 0 1
= 1 1 0 0 0 1 0 1

a) 173_{10} = 10101101 $_2$

b) $CD73_{16}$ = 1100 1101 0111 0011 $_2$

c) 110100101_2 = 645 $_8$

A 10
B 11
C 12
D 13
E 14
F 15

Question 2 (6 Marks)

Each of the binary numbers listed below are 8-bit 2's complement numbers. Perform the additions shown and set the flags as appropriate.

Show the result of the operation in both binary and decimal.

Operation	Answer	C	V	N	Z
00111001 + 11000100	= 11111101 $_2$ = ~~253~~ -3 $_{10}$	0	0	1	0
10110101 + 10101011	= 0110 0000 $_2$ = 96 $_{10}$	1	1	0	0

carry overflow negative zero

1 01100000

Question 3 (3 Marks)

For the IEEE single precision floating point representation of -17.321 complete the following:

The value of the sign bit : 1
The value stored for the exponent (in decimal) :
The stored mantissa (in binary, do not show trailing 0's) :

-17 = X11101111 X

0131748

Question 4 (7 Marks)

Consider the "Foo" representation of 8-bit signed numbers:

nnnnSnnn
(bit positions 4, 3, 2, 1, 0)

Where S denotes the sign bit, and n represents the binary value of the number.

Like signed-magnitude, this representation has two values for 0, and can represent the numbers from –127 to +127.

The table below shows some example numbers:

Decimal Value	"Foo" representation
107_{10}	11010011_2
-121_{10}	11111001_2

binary
0 1101011
1 1111001

Write a sequence of instructions that converts "Foo" numbers into 8-bit signed magnitude numbers.

You should assume that the number to convert is in ACCA, and the signed magnitude number should be in ACCA when your instructions are complete.

```
psha   /saves a on stack
lsra
lsra
lsra
lsra   /shifts register to the sign bit
aslb   /shifts carry bit to b accumulator
tsx    / x points to a on stack
lda 0,x  /reloads acca
ldy #0
asla   / shift a register right
aslb
asla
aslb
asla
aslb
asla
aslb
asla   / skips sign bit
asla
aslb
```

close, but you need rolb

you also end up w/ sign bit in the wrong ~~spot~~ place.

3/7.


```
asla
aslb
asla
aslb
pshb
pula
end
```

0131748

0.5

fetch
decode
execute
} for every command

subd 0,x

0 ↓ FF

1. fetch OP code
2. fetch operands (in this case fetch $00
3. add $00 to IX
4. fetch IX
5. fetch (IX+1)
6. subtract

ON FINAL

~~X~~ since 0,X ~~ea~~ the 0 can vary, it must be fetched

ind,X => A3 6 2 ← needs two bytes out of memory (2 fetches)

ind,Y => 18a3 7 3 ← needs three bytes out of memory since opcode is 2 bytes

18a3

Question 5 (3 Marks)

The instruction SUBD 0,X takes 6 clock cycles to execute. Describe what is happening at each clock cycle.

Ind,x

Cycle	Operation
1	op code instruction
2	Retrieves High byte of address - Read
3	Retrieves low byte of address - Read
4	Retrieves low byte of value
5	Retrieves high byte of value
6	write cycle

Question 6 (2 Marks)

a) Why are instructions using Direct Mode addressing (also called Zero Page addressing) 1 clock cycle faster than the same instruction using Extended Mode addressing?

Because the direct mode addressing only retrieves one bytesof data, whereas the extended mode retrieves 2 bytes of data.

0131748

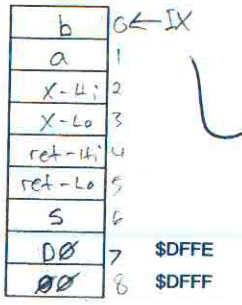
Question 7 (6 Marks)

```
        org   $D000
str1    fcc   "This is my string"
        fcb   $00

        org   $C000
        lds   #$DFFF
        ldx   #str1
        pshx
        ldaa  #'s'
        psha
        jsr   chcnt
        ins
        ins
        ins
        stop

chcnt   pshx
        psha
        pshb
        tsx
        ldy   #0          // IY is our counter
        ldab  6,x         // ACCB is the character we want to match
        ldx   7,x         // IX is our string pointer
loop    ldaa  0,x
        beq   chend
        inx
        cba
        bne   loop
        iny               // if A=B incIY
        bra   loop
chend   pulb
        pula
        pulx
        rts
        end
```

A) Draw the stack frame after the TSX instruction is executed.



a = T
ny = 0
b = 's'
x = str pointer

B) Fill in the appropriate offsets so the parameters are loaded correctly

C) Describe what this function does and what value it returns for the data shown above

This function counts the number of 's' characters in the desired string, and returns the value in the IY register.

IY = 3 after execution

0131748