Gebze Technical University Department of Computer Engineering CSE 101 – Introduction to Computer Engineering HW #1 Student 1801042669

Buğra Eren Yılmaz

8A9	1-a)	1000 1010 1001
EF3	1-b)	1110 1111 0011
0001 1110 0001	1-c)	1E1
1111 1110 1101 1011	1-d)	FEDB

2) -Approach to the problem (answer:“Computer”)

If we tried to decipher converted message digit by digit we wouldn't got a meaningful output from message.

For example, the first digit of converted message is "4" which corresponds in ASCII is "EOT(End Of Transmission)".

But if we take the first two digits of converted message, in this case "43", we will get "C" character from ASCII table which is a meaningful letter for us humans.

-Solution steps

2-a) We divide the converted string into sub-strings with spaces for every 2 digits. We get...
43 6F 6D 70 75 74 65 72

2-b) After rearrangement of the converted string, we will try to do reverse engineering and find the corresponds of hexadecimal numbers for ASCII system.
43 is “C” character, 6F is “o” character, 6D is “m” character...

After converting these hexadecimals to ASCII we get “Computer”.

3)

a	b
11111 (carry)	1 1 (carry)
00101 (5)	00101 (5)
+ 11111 (-1)	+ 10101 (-11)
00100 (4)	11010 (-6)

Because we are using 5-bits to store our data, let's assume in our memory, there is no memory cell available for another assignment. Due to that we eliminate the last carry of operation a. Both operations has no overflows.

4)

a	b	c
01001011	01001011	01001011
10101011	10101011	10101011
and	or	xor
00001011	11101011	11100000

5)

a) 7123

```
OR R1,R2,R3
```

b) 2BCD

```
LOAD RB, 0xCD
```

6)

```
LOAD R1, [0xA0]          ;Register for A0
LOAD R2, [0xA1]          ;Register for A1
LOAD R3, 11110000b       ;Masking sieve for first four bits
LOAD R4, 00001111b       ;Masking sieve for last four bits
AND R5,R1,R3             ;Masking A0's first four bits
AND R6,R2,R4             ;Masking A1's last four bits
XOR R7,R6,R5             ;Adding them together
STORE R7,[0xA2]          ;Store the result in A2
```