URBAN EDGE

Lab-3- Programs on Arithmatic and Logical Instructions

① Write an assembly language program to implement division by repetitive subtraction.

```
    AREA RESET, DATA, READONLY
    EXPORT __Vectors
__Vectors
    DCD 0x10001000
    DCD Reset_Handler
Reset_Handler
    LDR R0, =DEST1
    LDR R1, =DEST2
    MOV R2, #0
    LDR R3, =QUO
    LDR R4, =REM
    LDR R5, [R0]
    LDR R6, [R1]
LOOP  SUB R5, R5, R6
      ADD R2, R2, #1
      CMP R5, R6
      BHS LOOP
      STR R2, [R3]
      STR R5, [R4]
STOP B STOP
DEST1 DCD 46
DEST2 DCD 9
      AREA data, DATA, READWRITE
QUO DCD
REM DCD 0
    END
```

URBAN EDGE

OUTPUT

Final output

Memory

0x10000000: 05 00 00 01 00 00 00

Register

R0 → 0x2C → 0x2C

R1 → 0x30 → 0x30

R2 → 0x00 → 0x05

R3 → 0x10000000 → 0x10000000

R4 → 0x10000004 → 0x10000004

R5 → 0x2F → 0x0000001

R6 → 0x09 → 0x0000009

URBAN EDGE

(2) Find the sum of n natural numbers

```
        AREA RESET, DATA, READONLY
        EXPORT __Vectors
__Vectors
        DCD 0x10001000
        DCD Reset_Handler
        ALIGN
        AREA mycode, CODE, READONLY
ENTRY
        EXPORT mycode, CODE, READONLY Reset_Handler
Reset_Handler
        LDR R0, =NUM1
        LDR R1, [R0]
        LDR R7, =DEST
        MLA R1, R1, R1, R1
        MOV R2, #2
        MOV R3, #0
LOOP    SUB R1, R1, R2
        ADD R3, R3, #1
        CMP R1, R2
        BHS LOOP
        STR R3, [R7]
STOP    B STOP
NUM     DCD 5
        AREA data, DATA, READWRITE
DEST    DCD 0
        END
```

URBAN EDGE

Output

R0	0x2C	⟶	0x2C
R1	0x1E	⟶	0x00
R2	0x02	⟶	0x02
R3	0x00	⟶	0x0F
R7	0x10000000	⟶	0x10000000

Memory

0x1000000 - 0F 00 00 00 15 00 00 00

③ Write an assembly language ~~problem~~ program to find GCD and LCM of two 8 bits.

```
        AREA RESET, DATA, READONLY
        EXPORT __Vectors
__Vectors
        DCD 0x10001000
        DCD Reset_Handler
        ALIGN
        AREA mycode, CODE, READONLY
ENTRY
        EXPORT Reset_Handler
Reset_Handler
        LDR R0, =NUM1
        LDRB R1, [R0]
        LDR R0, =NUM2
        LDRB R2, [R0]
        CMP R1, R2
        BHI MSC
        MOV R3, R1
        MOV R1, R2
        MOV R2, R3
MSC     MOV R3, R2
FIX     MOV R4, R1
        MOV R5, R2
DIV     SUBS R4, R4, R3
        CMP R4, #0x0
        BMI RES
        BHI DIV
        BEQ DIV2
RES     SUB R3, R3#1
        BAL FIX
DIV2    SUBS R5, R5, R3
        CMP R5, #0x0
        BMI RES2
```

```
        BHI  DIV2
        BEQ  FIN

RES2    SUB  R3, R3, #1
        BAL  FIX

FIN     LDR  R5, =GCD
        STR  R3, [R5]
        MUL  R6, R1, R2
        MOV  R7, #0x0

DI      SUB  R6, R6, R3
        CMP  R6, #0x0
        ADD  R7, R7, #1
        BEQ  FINLCM
        BHI  DI

FINLCM  LDR  R8, =LCM
        STR  R7, [R8]

STOP    B STOP
NUM1    DCD 0x4
NUM2    DCD 0x2
        AREA data, DATA, READWRITE
GCD     DCD 0
LCM     DCD 0
        END
```

Output

R5: 0x10000000 → 02000000
R8: 0x10000004 → 04000000

R0 → 0x00 → 0x6C
R1 → 0x00 → 0x04
R2 → 0x00 → 0x02
R3 → 0x00 → 0x02
R7 → 0x00 → 0x04
R5 → 0x10000000
R6 → 0x10000004

4) WAP to convert 2 digit hexadecimal to ascii

```
AREA RESET, DATA, READONLY
EXPORT __Vectors
__Vectors
DCD 0x10001000
DCD Reset_Handler
ALIGN
AREA mycode, CODE, READONLY
ENTRY
EXPORT Reset_Handler
Reset_Handler
  LDR R0, =Num1
  LDR R1, [R0]
  AND R2, R1, #0x0F
  MOV R3, R1, LSR, #4
   cmp  R2, #0x09
   BLS DOWN
    Add  R2,R2, #0x09
DOWN ADD R2,R2, #0x30
second  cmp R3,#0x09
      BLS Down2
      Add  R3, R3,#0x07
Down2  ADD R3, R3, #0x30
STORE  LDR R4, =DEST0
       STR R2,[R4]
       LDR  R5, =DEST10
       STR  R3,[R5]
STOP B STOP
Num1 DCD 0x3A
   AREA data, DATA, READWRITE
DEST0 DCD 0
DEST10 DCD 0
  END
```

URBAN EDGE

(5) Output

R0 → 0x00000000 → 0x38
R1 → 0x00000080 → 0x3A
R2 → 0x00000000 → 0x41
R3 → 0x00000000 → 0x33
R4 → 0x00000000 → 0x10000000
R5 → 0x 00000000 → 0x10000004

0x10000000
↓
41 00 00 00 33 000000

5) WAP to convert 32 bit BCD number in unpacked to packed form

```
    AREA RESET, DATA, READONLY
    EXPORT __Vectors
__Vectors
    DCD 0x10001000
    DCD Reset_Handler
    ALIGN
    AREA mycode, CODE, READONLY
ENTRY
    EXPORT Reset_Handler
Reset_Handler
    LDR R0, =NUM1
    LDR R1, [R0]
    LDR R0, =NUM2
    LDR R2, [R0]
    MOV R3, R2, LSL # 4
    ADDS R4, R1, R3
    LDR R5, =DEST
    STR R4, [R5]
STOP B STOP
NUM1 DCD 0x1
NUM2 DCD 0x2
    AREA data, DATA, READWRITE
    DEST DCD 0
    END
```

Output: R0: 0x00000000 → 0x20

R1: 0x00000000 → 0x01

R2: 0x00000000 → 0x02

R3: 0x00 → 0x20

R4: 0x00 → 0x21

R5: 0x10000000

0x10.00.0000 : 21 00 00 00