ECEN 350 ARM Lab 5

Question 1:

Order in address,opcode, and disassembly

EL3:0x0000000080001278 :F0000100 ADRP x0,{pc}+0x23000 ; 0x80024278

EL3:0x000000008000127C :910B4000 ADD x0,x0,#0x2d0

EL3:0x0000000080001280 :F9400FA1 LDR x1,[x29,#0x18]

EL3:0x0000000080001284 :94000A55 BL printf ; 0x80003BD8

Question 2:

0x80003BD8

Question 3:

x0=0x800242D0

x1=0x8

Question 4:

Result of test(3,5) = %d\n

The result of the final value of the function above ﻿﻿﻿﻿﻿﻿i﻿﻿s stored in x0.

Question 5:

Printf uses the following to complete it's task:

EL3:0x0000000080001278 : ADRP x0,{pc}+0x23000 ; 0x80024278

EL3:0x000000008000127C : ADD x0,x0,#0x2d0

EL3:0x0000000080001280 : LDR x1,[x29,#0x18]

EL3:0x0000000080001284 : BL printf ; 0x80003BD8

The effective address of x0 is taken, then several bits are shifted in. The second argument is stored in x1. Then, a branch link printf is done.

Also it is worth mentioning that when the function is called, the following instructions are done:

EL3:0x0000000080001268 : MOV x1,#5

EL3:0x000000008000126C : MOV x0,#3

EL3:0x0000000080001270 : BL test ; 0x80001294

EL3:0x0000000080001274 : STR x0,[x29,#0x18]

With multiple arguments, they are loaded with MOV into x0,x1.....

Question 6:

.section .text

.globl test2

test2:

sub X0, X0, X1//x0=x0-x1

br X30

Question 7:

main:0x80001260

test:0x800012A4

test2:0x800012AC

Question 8:

main:0x80001260

test:0x800012A4

test2:0x800012AC

Question 9:

They are the same because of the way the main() function is ordered. ﻿

Question 10:

.section .text

.globl test2

test2:

sub X0, X0, X1 //subtracts second input from first input

mov x1,x0 //moves x0 into x1

ADR x0, mystring //gets the effective address of mystring

BL printf //branch link printf

br x30 //exit to main

.section .data

mystring: .asciz "Result of test2(6, 5) = %d\n"

//Working code without the use of stack