# CPE/EE 323 Introduction to Embedded Computer Systems
# Homework I

| 1(25) | 2(25) | 3(25) | 4 (25) | Total |
|---|---|---|---|---|
| | | | | |

**Problem #1 (25 points) Address Space, Memory**
Consider a hypothetical 24-bit processor called HYP24 with all registers, including PC and SP, being 24 bits long. The smallest addressable unit in memory is an 8-bit byte.
**A. (4 points)** What is the size of HYP24's address space in bytes and KB? How many address lines does HYP24 require?

Address space: _____Bytes Address space: _____ KB (KiloBytes).
Address bus lines: _____
**B. (6 points)** Assume that first quarter of the address space is dedicated for HYP24's RAM memory and the upper half of the address space is reserved for HYP24's Flash memory. Give address ranges for the RAM and Flash memories. Fill in the table below. What is the size of the RAM memory and the Flash memory?

| | Start byte address | End byte address |
|---|---|---|
| RAM memory | | |
| Flash memory | | |

RAM memory size [Bytes/KB]:_____
Flash memory size [bytes/KB]:_____

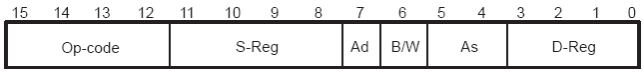
The MSP430F20x is a microcontroller with 64 KB of address space divided between code memory (flash), RAM memory, and input/output peripherals. It has 1,024 Bytes of RAM memory starting at the address 0x0200, and 256 Bytes of address space reserved for special purpose registers and 8-bit input/output peripherals (starting at the address 0x0000) followed by 256 Bytes reserved for 16-bit input/output peripherals. The flash memory of 8 KB resides at the top of address space (highest addresses in the address space).
**C. (8 points)** Determine the address map by filling in the following table.

| Address | Address [hexadecimal] | Sections in address space |
|---|---|---|
| Last Flash address | | Flash Memory |
| First Flash address | | |
| Last RAM address | | RAM Memory |
| First RAM address | | |
| Last I/O address (16-bit per.) | | I/O address space |
| First I/O address (16-bit per.) | | |
| Last I/O address (8-bit per.) | | I/O address space |
| First I/O address (8-bit per.) | | |

**D. (7 points)** What is the program stack (what is it, where is it located, and how we deal with it)? What is the maximum stack size in the MSP430Fx described above? What should be the initial value of SP?

**Problem #2 (25 points) MSP430 Addressing Modes, Instruction Encoding**

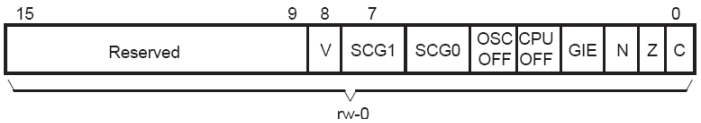
Consider the following instructions given in the table below. For each instruction determine its length (in words), the instruction words (in hexadecimal), source operand addressing mode, and the content of register R7 after execution of each instruction. Fill in the empty cells in the table. The initial content of memory is given below. Initial value of registers R5, R6, and R7 is as follows: R5=0xF002, R6=0xF00A, R7=0xFF88. Assume the starting conditions are the same for each question (i.e., always start from initial conditions in memory) and given register values. The format of the first word of double-operand instructions is shown below. (Note: Op-code for MOV is 0100).




| | Instr. Address | Instruction | Instr. Length [words] | Instruction Word(s) [hex] | Source Operand Addressing Mode | R7=? [HEX] |
|---|---|---|---|---|---|---|
| **(i)** | **0x1116** | **MOV R5, R7** | **1** | **0x4507** | **Register** | **0xF002** |
| **(ii)** | **0x1116** | **MOV.B R5, R7** | **1** | **0x4547** | **Register** | **0x0002** |
| (a) | 0x1116 | MOV 6(R5), R7 | | | | |
| (b) | 0x1116 | MOV.B 3(R5), R7 | | | | |
| (c) | 0x1116 | MOV.B -1(R6), R7 | | | | |
| (d) | 0x1116 | MOV EDE, R7 | | | | |
| (e) | 0x1116 | MOV.B TONI, R7 | | | | |
| (f) | 0x1116 | MOV &EDE, R7 | | | | |
| (g) | 0x1116 | MOV.B @R6, R7 | | | | |
| (h) | 0x1116 | MOV @R6+, R7 | | | | |
| (i) | 0x1116 | MOV #41, R7 | | | | |
| (j) | 0x1116 | MOV.B #27, R7 | | | | |

| Label | Address [hex] | Memory[15:0] [hex] |
|---|---|---|
| | 0xF000 | 0x0504 |
| | 0xF002 | 0xFFEE |
| TONI | 0xF004 | 0xCC06 |
| | 0xF006 | 0x3304 |
| | 0xF008 | 0xF014 |
| | 0xF00A | 0x2244 |
| EDE | 0xF00C | 0xABBA |
| | 0xF00E | 0xEFDD |

**Problem #3 (25 points) MSP430 Instructions, Addressing Modes**
Consider the following instructions given in the table below. For each instruction determine addressing modes of the source and destination operands, source and destination addresses, and the result of the operation. Fill in the empty cells in the table. The initial content of memory is given in the table. The initial value of registers R2, R5, and R6 is as follows: SR=R2=0x0000 (V=0, N=0, Z=0, C=0), R5=0x0403, R6=0xC006. <u>Assume the starting conditions are the same for each question (i.e., always start from initial conditions in memory) and given register values.</u>




| Label | Address [hex] | Memory[15:0] [hex] |
|---|---|---|
| | 0x0400 | 0xFEEE |
| | 0x0402 | 0xA000 |
| EDE | 0x0404 | 0xA4BC |

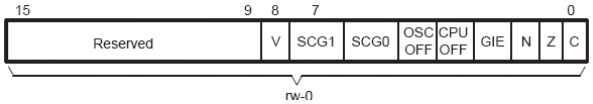
| Label | Address [hex] | Memory[15:0] [hex] |
|---|---|---|
| | 0xC000 | 0x0504 |
| | 0xC002 | 0xFEEE |
| TONI | 0xC004 | 0xA8FA |
| | 0xC006 | 0x33F4 |
| | 0xC008 | 0xF014 |
| DEN | 0xC00A | 0x2244 |
| | 0xC00C | 0xCDDA |

| | **Instruction** | **Instr. Size in Words** | **Source Operand Addressing Mode** | **Destination Operand Addressing Mode** | **Source Address** | **Dest. Address** | **Result (content of a memory location or a destination register; and new value of flags (C,V,Z, and N).** |
|---|---|---|---|---|---|---|---|
| (a) | ADD.B &TONI, R6 | | | | | | |
| (b) | SUBC TONI, -3(R5) | | | | | | |
| (c) | RRC.B @R5+ | | | | | | |
| (d) | AND.W #0xAA55, EDE | | | | | | |

Notes of setting flags: All instructions set N and Z flags as usual. Specific details for C and V are as follows: RRC (V=0, C is loaded with the shifted out bit).

**Problem #4 (25 points) MSP430 Instructions**

Consider the following instructions given in the table below. For each instruction determine changes in registers after its execution. Fill in the empty cells in the table. Initial value of registers R2, R5, and R7 is as follows: R2=0x0007 (Status register), R6=0xBB66, R7=0x40A9. Assume the starting conditions are the same for each instruction in the table (i.e., always start from the initial conditions in registers). Note: Format of the register R2 is shown below. For a detailed description of the instructions use the 5xx family user guide.




| Instruction | R7=0x???? | V | N | Z | C |
|---|---|---|---|---|---|
| ADD.B R6, R7 | **0x000F** | **0** | **0** | **0** | **1** |
| ADD R6, R7 | | | | | |
| ADDC R6, R7 | | | | | |
| SUB.B R6, R7 | | | | | |
| SUBC R6, R7 | | | | | |
| CMP.B R6, R7 | | | | | |
| CMP R6, R7 | | | | | |
| BIT R6, R7 | | | | | |
| BIC R6, R7 | | | | | |
| BIS R6, R7 | | | | | |
| AND R6, R7 | | | | | |
| XOR.B R6, R7 | | | | | |
| SWPB R7 | | | | | |
| RRC.B R7 | | | | | |
| RRC R7 | | | | | |
| RRA.B R7 | | | | | |
| RRA R7 | | | | | |