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?

$2^{24} = 16,777,216$ bits $/ 8 = 2,097,152$ B $/ 1024 = 2048$ KB

Address space: 2,097,152 Bytes Address space: 2048 KB (KiloBytes).

Address bus lines: 21 0001 1111 1111 1111 1111 1111

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	0000 0000 0000 0000 0000 0000	0011 1111 1111 1111 1111 1111
Flash memory	0100 0000 0000 0000 0000 0000	FFFFFF

RAM memory size [Bytes/KB]: 512

Flash memory size [bytes/KB]: 1024

2^{24}

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	0xFFFF	Flash Memory
First Flash address	0xE000	
Last RAM address	0x05FF	RAM Memory
First RAM address	0x0200	
Last I/O address (16-bit per.)	0x01FF	I/O address space
First I/O address (16-bit per.)	0x0100	
Last I/O address (8-bit per.)	0x00FF	I/O address space
First I/O address (8-bit per.)	0x0000	

0x0600

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?

The program stack, at the top of RAM is an area of the stack to temporarily store and preserve return addresses, procedure arguments, memory data, flags and registers.

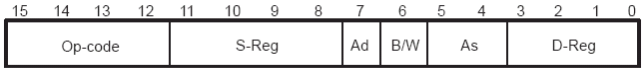
- Max stack size : 0x0200 -> 0x05FF
- Initial SP value : 0x0200

* moves down in memory

Max stack size = 1024 B

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).



F009 09

	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	2	0x4517 0x0006	Indexed	0xF014
(b)	0x1116	MOV.B 3(R5), R7	2	0x4557 0x0003	Indexed	0x0006
(c)	0x1116	MOV.B -1(R6), R7	2	0x4657 0xFFFF	Indexed	0x0014
(d)	0x1116	MOV EDE, R7	2	0x4017 0xDEF4	Symbolic	0xABBA
(e)	0x1116	MOV.B TONI, R7	2	0x4057 0xDEED	Symbolic	0xCC06
(f)	0x1116	MOV &EDE, R7	2	0x4217 0xF00C	Absolute	0xABBA
(g)	0x1116	MOV.B @R6, R7	1	0x4667	Indirect	0x0044
(h)	0x1116	MOV @R6+, R7	1	0x4637	Indirect auto	0x2244
(i)	0x1116	MOV #41, R7	2	0x4237 0x0029	Immediate	0x0029
(j)	0x1116	MOV.B #27, R7	2	0x4277 0x001B	Immediate	0x001B

only after instruction

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

a) R5+6 = 0xF008; M[0xF008] = 0xF014 ∴ R7=0xF14

0100_0101_0|0|0|1_0111
Op-code R5 Ad B/w As R7

b) R5+3 = 0x0005 R7 ← M[0005]

0|1|0|1

F004
R5 F002 + F005 → second word in byte
F006

0100 0000 0|1|0 0 0000
Op-code Source Ad B/w As Destination

c) R6-1 R7 ← M[F009]

0100_0110_0101_0111

d) R7 ← M[EDE]
R7 ← M[F00C]
0100_0000_0001_0111

e) R7 ← M[F004] → R7 ← 0006

0100_0000_0101_0111

f) R7 ← M[EDE]
0100_0000_0001_0111

g) MOV.B @R6, R7 0xF00A
0100_6_0110_0111
R7 ← M[R6]
R7 ← 2244
R7 ← 0x0044

h) MOV @R6+, R7 R7 ← M[R6]
0100_0010_0011_0111 R6 → F00C

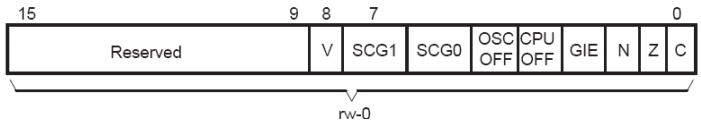
i) MOV #41, R7 R7 ← 41
0100_0000_0011_0111 ⇒ 4037

MOV.B #27, R7 R7 ← 27
0100 0000 0111 0111

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. Assume the starting conditions are the same for each question (i.e., always start from initial conditions in memory) and given register values.

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



R2 = 0x0000
R5 = 0x0403
R6 = 0xC006
0000
V N Z C

	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 src dst R6 ← M[TONI] R6 ← FA + 06	2	Absolute	Register	0xC004	0x0006	C V Z N 1 0 1 0 R6 = 0x0000
(b)	SUBC TONI, -3(R5) src dst R5 0x0400 0x400 ←	3	Symbolic	Indexed	0xC004	0x0400	C V Z N 1 0 0 0 55F4
(c)	RRC.B @R5+	1	Indirect Auto increment	N/A	0x0403	N/A	C V Z N 1 0 0 0 [R5] = 0x0002 R5 = 0x0404
(d)	AND.W #0xAA55, EDE	3	Immediate mode	Symbolic	0x0000	0x0404	C V Z N 0 0 0 1 [EDE] = 0xA014

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).

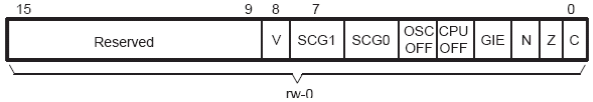
1111 1010
0000 0110
0000 0000
100
V=0
C=1
N=0
Z=0

0x0003
0000 0000 0000 0011 0
0000 0000 0000 0001 1

AA55
A4BC
1010 1010 0101 0101
1010 0100 1011 1100
1010 0000 0001 0100
A 0 1 4

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.



0000 0000 0000 0111
V N Z C

R6=0xBB66, R7=0x40A9.

	Instruction		R7=0x????	V	N	Z	C
	ADD.B	R6, R7	**0x000F**	**0**	**0**	**0**	**1**
a	ADD	R6, R7	0xFC0F	0	1	0	0
b	ADDC	R6, R7	FC10	0	1	0	0
c	SUB.B	R6, R7	0x0043	0	0	0	1
d	SUBC	R6, R7	0x8543	0	1	0	0
e	CMP.B	R6, R7	0x40A9	0	0	0	1
f	CMP	R6, R7	0x40A9	0	0	0	0
g	BIT	R6, R7	0x40A9	0	0	0	0
h	BIC	R6, R7	0x0088	0	1	1	1
i	BIS	R6, R7	FBEF	0	1	1	1
j	AND	R6, R7	0x0020	0	0	0	0
k	XOR.B	R6, R7	0x00CF	0	1	0	0
l	SWPB	R7	A940	0	1	1	1
m	RRC.B	R7	0x0004	0	1	0	1
n	RRC	R7	0xA054	0	1	0	1
o	RRA.B	R7	0x00D4	0	1	0	1
p	RRA	R7	0x2054	0	0	0	1

BB66
+40A9
FC0F
+ 1

R7-R6 A9-66 = 0x0043

dst - src + C

0100 0100 1001 1001
0100 0000 1010 1001
0100 0000 1000 1081

1011 1011 0110 0110
0100 0000 1010 1001
1111 1001 1110 1111

1011 1011 0110 0110
0100 0000 1010 1001
0000 0000 0010 0000

1010 1001
0110 0110
1100 1111

A9 1010 1001 => 1101 0100 1

0100 0000 1010 1001
1010 0000 0101 0100 1

A9 1010 1001 1101 0100 1

40A9 0 0100 0000 1010 1001

0010 0000 0101 0100 1