# EXAM SEPTEMBER 12, 2012. SOLUTIONS

**Problem 1**. Answer the following questions:

1.1. What is an embedded system?
*ANSWER: An embedded system contains tightly coupled hardware and software components to perform a single function and forms part of a larger system*

1.2. List five attributes of embedded systems.
*ANSWER: (a) It forms part of a larger system; (b) is not intended to be independently programmable by the user, (c) it is expected to work with minimal or no human interaction; (d) It is reactive; (e) heavily constrained; (f) Mostly operating in real time*

1.3. What are the fundamental component of an embedded system?
*ANSWER: The hardware and software components. Hardware components: CPU, Memory, hardware interconnections, communication ports, etc. Software components: the kernel, tasks, task services.*

1.4. What is a "word" in digital systems? *ANSWER: Is an ordered sequence or permutation of m bits*

1.5. What is named the "least significant bit" and what the "most significant bit " in a word? *ANSWER: least significant bit of a word is the bit at the rightmost position; most significant bit is the one at the leftmost position.*

1.6. How many different words can be written using N bits? *ANSWER:* $2^N$

1.7. What is a bit? *ANSWER: a 0 or a 1*

1.8. Define the terms "nibble", "byte", "double word" and "quad". *ANSWER: nibble = 4 bits; byte = 8 bits; double word = 32 bits; quad = 64 bits*

1.9. What power of two means Tera? *ANSWER:* $2^{40}$

1.10. How many digits are there in a positional system base H? *ANSWER: H*

1.11. In a positional system base r, what is the power series for $a_{n-1}a_{n-2}\dots a_1a_0.a_{-1}a_{-2}a_{-3}$?
*ANSWER:* $a_{n-1}r^{n-1} + a_{n-2}r^{n-2}\dots + a_1r + a_0 + a_{-1}r^{-1} + a_{-2}r^{-2} + a_{-3}r^{-3}$

1.12. What is the hex notation for the maximum word that can be written using 15 bits? *ANSWER: 111 1111 1111 1111b = 7FFFh*

1.13. How do you write $16^4$ in hexadecimal base? *ANSWER: 10000h*

1.14. What are the hardware and software architecture components of a CPU?
*ANSWER: Hardware components: ALU (Arithmetic Logic Unit), CU (Control Unit), Registers, BIL (Bus Interface Logic unit); Software components: Instruction set and addressing modes*

1.15. What does CPU stands for?*ANSWER: Central Processing Unit*

1.16. List the four basic special purpose registers that we expect to find in any CPU.
*ANSWER: Instruction Register (IR), Program Counter (PC) or Instruction Pointer (IP), Status Register (SR) and Stack Pointer (SP)*

1.17. What are the main flags that we expect to find in any status register?
*ANSWER: Carry C, Zero Flag Z, Negative or Sign flag N, Overflow flag (V or O) and Interrupt Enable flag (IE, GIE, I)*

1.18. Which is the CPU component that determines the number of bits attached to the name or characteristic of a microcontroller? (In other words, what do we mean by terms like "8-bit microcontroller"?) *ANSWER: The ALU (8-bit controller means it has an 8-bit ALU)*

1.19. What are the phases in an instruction cycle? *ANSWER: Fetch, Decode and Execute*

1.20. Define "addressing mode". *ANSWER: Addressing mode is the way or syntax of used for an operand to define where the data to operate with is*

**Problem 2**. A CPU has four general purpose 8-bit registers ACC, GPR1, GPR2 and GPR3. In an arithmetic instruction, only the elements that are shown in Figure P. 2 take part. This is a system view, with 8-bit multiplexers, demultiplexers, and so on. Bits A6, A5, A4, A3, A2, A1, and A0 are part of the instruction word. The output X of the full adder is fed back to the input of the demultiplexer. Your task is to analyze four instructions determined by the instruction word and fill the table below. The operations are independent one of the other. The operation takes place when the system clock goes through a pulse. (You can use programming notation like A=A+B or any other you are used to)

The contents of the registers before each operation is as follows:

ACC = F2h, GPR1 = 35h, GPR2 = DBh, GPR3 = 76h.

The table lists six instructions and the first one is filled for your reference:

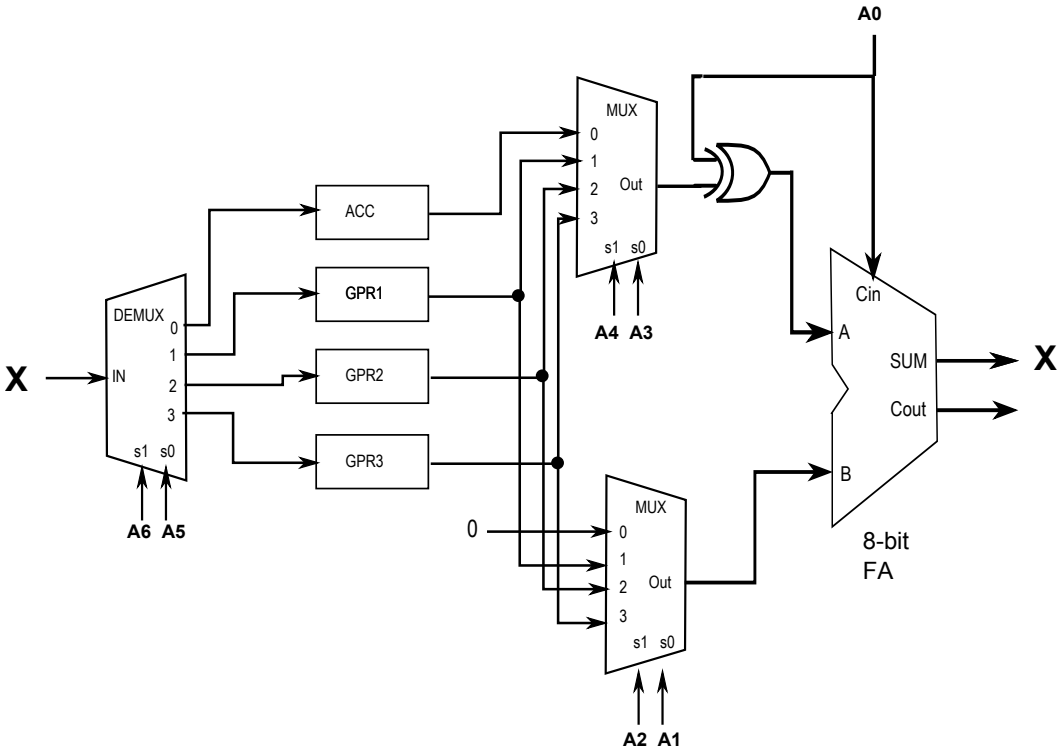


Figure P. 2: ALU and register portion of CPU for instruction analysis

| A6 | A5 | A4 | A3 | A2 | A1 | A0 | ACC | GPR1 | GPR2 | GPR3 | Operation/comment |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | F2h | 35h | DBh | 76h | Nothing happens, since ACC is copied onto itself. |
| 0 | 1 | 1 | 0 | 0 | 1 | 0 | | | | | |
| 1 | 1 | 0 | 0 | 0 | 0 | 1 | | | | | |
| 0 | 0 | 1 | 0 | 1 | 1 | 1 | | | | | |
| 1 | 0 | 1 | 1 | 1 | 0 | 0 | | | | | |

**SOLUTION:** We can solve this problem by looking at the meaning of the bits in the instruction word:

A0: X is the output of the full adder at the right (which forms part of the ALU). If A0=0, then X=A+B, where A and B are the full adder inputs. If A0=1, then X = B + A'+1, that is B-A or equivalently B plus two's complement of A.

A2-A1 and A4-A3 select what registers will be connected to inputs A and B of the full adder according to the following tables:

| A4 | A3 | A |
|---|---|---|
| 0 | 0 | ACC |
| 0 | 1 | GPR1 |
| 1 | 0 | GPR2 |
| 1 | 1 | GPR3 |

| A2 | A1 | B |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | GPR1 |
| 1 | 0 | GPR2 |
| 1 | 1 | GPR3 |

A6-A5: They control a de-multiplexer. The input X will be transferred to an output 0-3, selected by these bits (the table is like that for A4-A3). At the clock-pulse, this X value will be stored in the corresponding Register. Let's call this register C.

Accordingly:

A6A5 A4A3 A2A1 A0 = 01 10 01 0 means GPR1 = GPR2 + GPR1, that is, add contents of GPR1 and GPR2 and store result in GPR1. Since 35h + DBh = 110h, the carry out is 1, and GPR1 is now 10h.

A6A5 A4A3 A2A1 A0 = 11 00 00 1 means GPR3 = $\overline{\text{ACC}}$+ 1 + 0, or GPR3=0-ACC=-ACC. GPR3 is now the two's complement of the ACC contents, 0Eh.

We proceed similarly with other cases to fill the table:

| A6 | A5 | A4 | A3 | A2 | A1 | A0 | ACC | GPR1 | GPR2 | GPR3 | Operation/comment |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | F2h | 35h | DBh | 76h | Nothing happens, since ACC is copied onto itself. |
| 0 | 1 | 1 | 0 | 0 | 1 | 0 | F2h | 10h | DBh | 76h | GPR1=GPR2+GPR1 |
| 1 | 1 | 0 | 0 | 0 | 0 | 1 | F2h | 35h | DBh | 0Eh | GPR3 = 0-ACC (Two's complement of ACC) |
| 0 | 0 | 1 | 0 | 1 | 1 | 1 | 9Bh | 35h | DBh | 76h | ACC = GPR3 - GPR2 |
| 1 | 0 | 1 | 1 | 1 | 0 | 0 | F2h | 35h | 51h | 76h | GPR2 = GPR3+GPR2 |

**Problem 3**. Answer the following questions:

3.1. In a numerical system of base $r$, $19_r/3_r=7_r$. What is the value of $r$?

**Solution:** The operation is equivalent to $19_r=3_r \times 7_r$. Since digits 0 to 9 have the same value independently of the base, we can write $19_r = r+9 = 21$. Hence, $r = 12$.

3.2. In a numerical system of base $r$, the polynomial $x^2 - 11x + 22 = 0$ has the solutions 3 and 6. What is the base $r$ of the system?

**Solution:** From Algebra, the statement is equivalent to say that $(x^2 - 11x + 22)_r = (x-3)_r(x-6)_r$. Doing operations we arrive at 3+6=$11_r$=r+1, and (3)(6)=$22_r$=$2(11_r)$. In any case, $r = 8$.

3.3. The conversion of numbers containing fractional parts is most often approximate. To check this statement, convert 3.27 to binary with 6 bits only, and then convert your result back to decimal. What is the difference between the original decimal number and the retrieved one from the binary equivalent?

**Solution:** Integer part 3 = 11b. Now we convert 0.27 stopping at four bits:

0.27 X 2 = 0.54; 0.54 X 2 = 1.08; 0.08 X 2 =0.16; and 0.16 X 2 = 0.32.

Therefore, limited to four bits, we have 3.27 = 11.0100b. If we convert 11.0100B to base 10 we get 3.25. Hence, the error or difference is 0.02 (this called quantization error).

3.4. Using binary numbers, calculate 84 - 31 with normal subtraction and using two's complement addition.

**Solution:** Converting 84 and 31 we have 84 = 1010100B and 31 = 11111B. The normal subtraction yields:

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 84→ | 1 | 0 | 1 | 0 | 1 | 0 | 0 | − |
| 31→ | | | 1 | 1 | 1 | 1 | 1 | (−) |
| Borrow | 1 | 1 | 1 | 1 | 1 | 1 | | = |
| | 0 | 1 | 1 | 0 | 1 | 0 | 1 | |

0110101B=53, as it should be. Now, for the two's complement of 31 we must consider 7 bits, since 84 has this number of bits. 31=0011111B, and the two's complement is (31)=1100001B. We proceed now as:

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Carry | 1 | 0 | 0 | 0 | 0 | 0 | 0 | | (+) |
| 84→ | | 1 | 0 | 1 | 0 | 1 | 0 | 0 | + |
| (31)→ | | 1 | 1 | 0 | 0 | 0 | 0 | 1 | = |
| | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | |

Discarding the leftmost carry bit 1, we arrive at the same result.

**Problem 4**. When two binary numbers have the same number of bits N (left zeros if necessary), then the following "flags" are generated by addition or subtraction:
a) Carry C=1 if the operation generates a carry and 0 otherwise;
b) Zero Z=1 if the N bits of the result are 0 –any Carry excluded –, and 0 otherwise;
c) N, equal to the most significant bit of the result – Carry excluded –; and
d) V (overflow) =1 if

d.1) the addition of two numbers whose most significant bit are equal yields a result with different most significant bit; or

d.2) The subtraction A-B with A and B having different most significant bits, yields a result with the same significant bit as B.

Determine C, Z, N and V for the following operations for 12-bit data:

4.1. 72Ah + 8F1h

4.2. A4Bh + C20h

4.3. 123h - 432h (using two's complement)

4.4. D21h + 2DFh

4.5. C1Ah - DBh (using two's complement)

**Solution:** To make it clearer, the operations are repeated in binary numbers. It is a good practice to also do it in hex numbers without going through the conversion.

4.1
72Ah + 8F1h = 101Bh:

```
     011100101010   +
     100011110001   =
  1 ------------
    000000011011
```

C=1; Z=0; N=0; and V=0

4.2
A4Bh + C20h = 166Bh:

```
     101001001011   +
     110000100000   =
  1 ------------
    011001101011
```

C=1; Z=0; N=0; and V=1

4.3
123h - 432h done as
123h + BCEh = CF1h:

```
     000100100011   +
     101111001110   =
    ------------
     110011110001
```

C=0; Z=0; N=1; and V=0

4.4
D21h + 2DFh = 1000h:

```
     110100100001   +
     001011011111   =
  1 ------------
    000000000000
```

C=1; Z=1; N=0; and V=0

4.5
C1Ah - DBh done as
C1Ah + F25h = 1B3Fh

```
     110000011010   +
     111100100101   =
  1 ------------
    101100111111
```

C=1; Z=0; N=1; and V=0

**Problem 5**. Answer the following questions.

5.1. Explain the function of the registers PC, IR, SR and SP

**Solution:** PC holds the memory address of the next instruction to be fetched. IR holds the instruction word currently in the cycle. The bits in SR are the flags and other bits that indicate the state of the CPU. SP holds the memory address for the Top of Stack.

5.2. Explain the steps followed in each of the fetch-decode-execute phases of an instruction cycle indicating clearly the registers involved in the process and what is their role.

**Solution:** In the fetch phase, the control unit places the contents of the PC register in the Bus Interface Logic Unit and places the instruction in the Instruction Register IR.

In the decode phase, the contents of the IR is decoded by the CU to establish the operation to be executed and the addressing modes of the operands, so it can proceed accordingly. After this phase, the PC has been incremented to point to the next location after the instruction currently in the cycle. (That is, the next instruction for the following fetch phase)

In the execute phase, the CPU executes the instruction. The registers involved in this phase will depend on the instruction.