EE2016 Microprocessors Theory and Lab, Aug - Nov 2024.

Tutorial 3 (Classes on 26th & 27th of Aug and 2nd & 3rd of Sep, 2024)

Dr. R. Manivasakan, EE Dept, IIT Madras.

Portions: Wk5-Start ALU, shift register, flag register and their interconnection through address, control and data lines, ALU - components (logical & arithmetic operation engines), FPU, addressing modes (immediate, direct, indirect, register direct, register indirect), illustration of MOV in hardware, LDI ditto, assembly language CU, EUCI, PFCI, EUCL, PFCL, stack pointer, link register, EU & CU put together, AVR hardware architecture (Mazidi), RISC-style & CISC-style assembly programming.

1 Fill in the blanks

1. Instruction size is ..fixed..... (fixed / variable) in AVR.
2. The number of general purpose registers (GPRs) within the AVR processor is .....32.....
3. The D7 bit of SREG in AVR denotes .....global interuupt enable..... .
4. In mask ROM, the contents are programmed by the ..... (manufacturer / user).
5. In programmable ROM, the contents are programmed by the ..... (manufacturer / user).
6. In EPROM, selectively a byte ..... (can / cant) be erased and rewritten.
7. In Electronically Erasable PROM (EEPROM), selectively a byte ..... (can / cant) be erased and rewritten.
8. Flipflops are used to construct ..... (SRAM / DRAM).
9. Capacitors are used to construct ..... (SRAM / DRAM).
10. Volatile SRAM is made up of ..... (FFs / semiconductor capacitors) and is made to be non-volatile by using ..batteries..... .
11. The advantage(s) of DRAM over SRAM is (are) ...cheaper, power consumption less
12. The advantage(s) of SRAM over DRAM is (are) ...faster..... .
13. The pair R27 R26 hold ..16.. bit address in the main memory and could be used to uniquely pull one location out of ..... locations.2^16
14. Stack pointer (SPH & SPL) is located in ..... (GPR region or I/O registers area).
15. Enumerate the difference between IN and LDS instructions.
16. Instructions that affect all the six flag bits C, H, Z, S, V and N are ..... (ADD, AND, DEC, SUB, INC).
17. The difference between arithmetic shift left and logical shift left is (are) .....holds the sign bit..... .....adds zero to the empty bit.....
18. Every assembly instruction consist of two parts: first part is .....opcode..... and the second one being ..operand.....

19. The EUCI ..... (does / doesn't) change the sequence of program execution.
20. The PFCI ..... (does / doesn't) change the sequence of program execution.
21. The hardwired control unit consists of ....., ... and ..... control matrix, instruction decoder, pulse generator

2 Answer the following

1. **Research Problem (Optional):** Ref the paper "Fast Digital Convolutions using Bit-Shifts" by S S Chandra, for the following question: Ref the paper (uploaded in moodle) in which equ (2) is of centre of our problem. This is repeated here for convenience:

$$X(u) = \sum_{t=0}^{N-1} \left\langle x(t) \cdot 2^{ut} \right\rangle_m \tag{1}$$

   (a) Establish that the above equ can be implemented by shifting, modulo and addition operations. Refer paper (& any other sources) for further information. [Note multiplication is avoided here. It is also not repeated additions]. Evaluate the time complexity - number of clock cycles required.

   (b) Implement the computation of the above convolution using the conventional multiplication and sum. Evaluate the number of clock cycles required.

   (c) Compare the number of computations required in part (a) and part (b) and explain the difference.

2. This problem is about the addressing modes. Given the following:

   (a) Each mode field is 8 bits

   (b) Internal memory 32, 8-bit registers

   (c) 1 kbyte of main memory with word size 2 bytes

   compute,

   (a) how many distinct unique memory locations, each of the following five type of addressing, can address?
      i. immediate
      ii. direct
      iii. indirect
      iv. direct register
      v. indirect register

      i. NILL | 2^8
      ii. 2^8 | 2^16 [only dependent on word size]
      iii. 2^9 - 1 | 2^16 [only 512-memory among which one already used by some indirect reg.]
      iv. 32 | 2^8
      v. 2^8 | 2^16
      [generally, it should have been 2^5 but these 2^5 location have 8- bit address and these 8 bits can point to 2^8 unique loc]

   (b) the range of values of operands each of the above addressing could handle

   Present the answer in a tabular form.

3. The address bus of a computer has 16 address lines, $A_{15} - A_0$. The hexadecimal address assigned for a given device, is 7CA4.

   (a) Draw the address decoder (digital circuit). [Is it combinational circuit only or it is a combination of sequential & combinational logic?] 16 bit and gate

   (b) At a later point in time if the above address decoder is modified such that the device ignores lines A_8 and A_9, what are all the addresses to which this device will respond? [Mention the range of addresses]. 7CA4, 7DA4, 7EA4, 7CF4

4. **Floating Point Unit (FPU):** The intention of this problem is to understand the internal engines in FPU. Ref pp 327 - 342 in Stallings. In particular, the 4 steps for addition or subtraction given in p.334. Given the numbers $X = 0.037$ and $Y = 70045.467$

(a) How are the numbers X and Y are internally represented through IEEE 754 floating point representation in normalised form with precision of two decimal places.

(b) To compute the sum X+Y, follow the 4 steps given in p.334 and evaluate each step in detail.

(c) Throw light on specifically the role of shifters in this problem. How many places does the shifter need to shift (any word for that matter) in this problem?

5. Consider the AVR assembly program given in the class

LN	Instructions
1	LDI R4, 10
2	LDI R5, 20
3	LDI R0, 0
4	INC R4
5	MVI R1, R(R4)
6	ADD R0, R1
7	MOV R0, R2
8	CP R4, R5
9	BREQ STOP
10	JUMP LOOP
11	NOP

```
LDI R4, 10
LDI R5, 20
LDI R0, 0
MOV ZL, R4
MOV ZH, R0
LPM R1, Z+
ADD R0, R1
cp
```

The above program corresponds to an hypothetical processor in which the indirect address register (IAR) is used to implement the indirect addressing. Rewrite the above program corresponding to AVR (say 8 bit processor) - Use only the AVR assembly syntax. You MUST use pointer registers in GPR area. [In week 8 - 9, for the above task, rewrite the code, using ARM assembly syntax].

6. **RISC-style versus CISC-style Assembly Program (Optional)**: Problem is the addition of two numbers in two distinct memory locations in the main memory and storing the result in a third memory location. SRAM data memory locations in a RISC (AVR) processor is 0x00A0 and 0x00A1. Final storing location is 0x00A2. Store the carry (if at all, is generated), in 0x00A3. Write a RISC style assembly program (use AVR ISA). Similarly identify a CISC processor like intel and write a CISC style assembly program for the above task. Compare the above two in terms of the following

(a) How many clock cycles each of the above processors take?

(b) How much is the memory occupied by their respective assembly programs?

(c) Given the average frequency of an AVR processor being 16 MHz and that of intel processor being 3 GHz, compute the total time for the above program, in each of the above cases.

7. [**Dot Product Computation** - Hamachar Pr. 2.26] The dot-product computation is discussed in Section 2.12.1, Hamachar 6e. This type of computation can be used in the following signal-processing task. An input signal time sequence IN(0), IN(1), IN(2), IN(3), . . . , is processed by a 3-element weight vector (WT(0), WT(1), WT(2)) = (1/8, 1/4, 1/2) to produce an output signal time sequence OUT(0), OUT(1), OUT(2), OUT(3), . . . , as follows:

   OUT(0) = WT(0) × IN(0) + WT(1) × IN(1) + WT(2) × IN(2) OUT(1) = WT(0) × IN(1) + WT(1) × IN(2) + WT(2) × IN(3) OUT(2) = WT(0) × IN(2) + WT(1) × IN(3) + WT(2) × IN(4) OUT(3) = WT(0) × IN(3) + WT(1) × IN(4) + WT(2) × IN(5) .. .

   All signal and weight values are 32-bit signed numbers. The weights, inputs, and outputs, are stored in the memory starting at locations WT, IN, and OUT, respectively. Write a RISC-style program to calculate and store the output values for the first n outputs, where n is stored at location N.

   Hint: Arithmetic right shifts can be used to do the multiplications.

8. [**Hamachar, Pr. 9.21, Optional**] In Section 9.7 (Hamachar 6e), we used the practical-sized 32-bit IEEE standard format for floating- point numbers. Here, we use a shortened format that retains all the pertinent concepts, but is manageable for working through numerical exercises. Consider that floating-point numbers are represented in a 12-bit format as shown in Figure below. The scale factor has an implied base of 2 and a 5-bit, excess-15 exponent, with the two end values of 0 and 31 used to signify exact 0 and infinity, respectively. The 6-bit mantissa is normalized as in the IEEE format, with an implied 1 to the left of the binary point.

   (a) Represent the numbers +1.7, −0.012, +19, and 18 in this format.

   (b) What are the smallest and largest numbers representable in this format?

   (c) How does the range calculated in part (b) compare to the ranges of a 12-bit signed integer and a 12-bit signed fraction?

   (d) Perform Add, Subtract, Multiply, and Divide operations on the operands