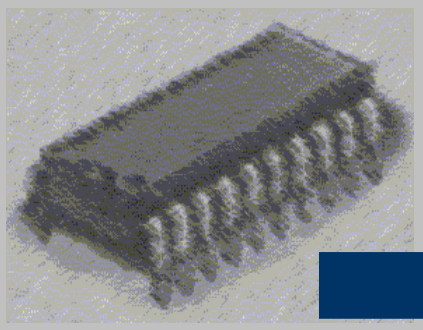
Exercises

Lesson 2. Instruction set design



Computer Structure and Organization

Graduate in Computer Sciences
Graduate in Computer Engineering

1. Lets a 32 bits word computer with a register file of 16 registers of 32 bits (.1 is PC and .2 SP) Memory capacity is 256 Mwords.

   Instruction set is a very orthogonal small one:

   - Move source, target
   - Add target, operand1, operand2

   Allowed addressing modes are: immediate, direct to register and direct to memory, relative to register, relative to index register with pre / post increment / decrement and indirect.

   Operation code and operands (with addressing mode information) are concatenated to build the machine code.

   Please, design instruction format for above architecture

---

2. Lets a 16 bits word computer with 16 registers in the register file. Please, design register-register instruction format by using expansion operation code that allow:

   - 15 instructions of 3 operands
   - 14 instructions of 2 operands
   - 31 instructions of 1 operand
   - 16 instructions of 0 operands

   Please, specify field lengths and the range for operation code values.

---

3. Lets a 16 bits word computer with a 32 register file that is able to execute next orthogonal instruction set:

   - Move source, target
   - Movec source, target, condition

- Moved source1, target1, source2, target2, condition
- Add operand1, operand2, target
- Sub operand1, operand2, target
- Mul operand1, operand2, target
- Div operand1, operand2, target
- And operand1, operand2, target
- Or operand1, operand2, target
- Xor operand1, operand2, target
- Shift source, target, type, counter

Where:

- Conditions are: C, NC, Z y NZ
- Addressing modes are: immediate, direct to register and relative to register.
- Allowed representation system for data are: unsigned, complement 2 and floating point.
- Same representation system required for all the operands in the same instruction

Please, design instruction set format.

---

4. Lets a 32 bits word computer with a 32-register file. Computer has a 60 instruction set and the allowed addressing modes are:
   - Direct
   - Indirect
   - Relative to base register
   - Relative to program counter

Please, design Register-Memory instruction formats only.

- Please, design an extended operation code instruction set for a 36 bits word computer in which we have:
- 7 instructions of two 15 bits fields and one of 3 bits
- 500 instructions of one field of 15 bits and another of 3 bits
- 50 instructions with no operands

---

5. Lets a 16 bits word computer with a 16-register file. Please, design register-register instruction formats by using expansion operation code technique with following requisites:
   - 15 instructions of 3 operands
   - 14 instructions of 2 operands
   - 31 instructions of 1 operand
   - 16 instructions of 0 operands

   Please, specify field length and the range for operation code values.

---

6. The same instruction set is implemented in two different computers with the same architecture two. Same program execution characteristics are:

	Clock cycles	Cycles per instruction (CPI)
Architecture 1	3	2
Architecture 2	4	1,5

   Please, indicate which is the quickest architecture of two above and how much?

7. Lets classify bellow architectures according to their instruction set:

Architecture					
Format	**0 operands**	**1 operand**	**2 operands**	**3 operands**	
Execution model	**Stack**	**Accumulator**	**Register-Register**	**Memory-Memory**	**Register-Register**

Instruction set	PUSH M
	POP M
	ADD
	SUB
	MUL
	DIV
		LOAD M
		STORE M
		ADD M
		SUB M
		MUL M
		DIV M
			LOAD X, M
			STORE M, X
			MOVE X, Y
			ADD X, Y
			SUB X, Y
			MUL X, Y
			DIV X, Y
				ADD M1, M2, M3
				SUB M1, M2, M3
				MUL M1, M2, M3
				DIV M1, M2, M3
					LOAD X, M
					STORE M, X
					MOVE X, Y
					ADD X, Y, Z
					SUB X, Y, Z
					MUL X, Y, Z
					DIV X, Y, Z

- Denoted M operands are memory positions and X, Y or Z operands are registers of the register file.
- 0 operands machine has an execution stack model that means operands must be removed from the top of the stack and the result is also stored on the top of the stack when using process instructions. PUSH and POP instruction are the only one in accessing to memory.
- 1 operand machine has an accumulator execution model. Process operations are performed with a memory operand and the implicit accumulator. LOAD M is used to load a value on the accumulator. STORE M is to store accumulator value on M memory position.
- 2 operands machine is a load-store machine. Operations are performed on some of the 16 registers of the file register. Transfer instructions move information between memory and registers (LOAD, STORE) or between register themselves (MOVE)
- 3 operand machines specify 3 operands per instruction. Two possibilities are proposed: memory-memory execution model and register-register execution model.
- Program bellow expression in each one of the above machines execution model (suppose all variables are in memory)

$$A = ((B + C) \cdot D) / (E – F \cdot G – H \cdot I)$$

8. Please, measure memry performance of propposed architectures above. Suppose:
    - Operation codes are 1 byte length
    - Memory addressing modes are 2 bytes length
    - Operands are 2 bytes length
    - Registers use 4 bits to be specified (16 registers)

Please, calculate the length of above programs and how many bytes are transferred across data bus.

Which one of the above programs is the most efficient according to executable code size only?

Please, calculate which is the most efficient according to required data bandwidth (code + data)