Chapter 1

Introduction

This Chapter gives you…

- System Software & Machine Architecture
- The Simplified Instructional Computer SIC and SIC/XE
- Traditional (CISC) Machines - Complex Instruction Set Computers
- RISC Machines - Reduced Instruction Set Computers

1.0 Introduction

The subject introduces the design and implementation of system software. Software is set of instructions or programs written to carry out certain task on digital computers. It is classified into system software and application software. System software consists of a variety of programs that support the operation of a computer. Application software focuses on an application or problem to be solved. System software consists of a variety of programs that support the operation of a computer. Examples for system software are Operating system, compiler, assembler, macro processor, loader or linker, debugger, text editor, database management systems (some of them) and, software engineering tools. These software's make it possible for the user to focus on an application or other problem to be solved, without needing to know the details of how the machine works internally.

1.1 System Software and Machine Architecture

One characteristic in which most system software differs from application software is machine dependency.

System software – support operation and use of computer. Application software - solution to a problem. Assembler translates mnemonic instructions into machine code. The instruction formats, addressing modes etc., are of direct concern in assembler design. Similarly, Compilers must generate machine language code, taking into account such hardware characteristics as the number and type of registers and the machine instructions available. Operating systems are directly concerned with the management of nearly all of the resources of a computing system.

There are aspects of system software that do not directly depend upon the type of computing system, general design and logic of an assembler, general design and logic of a compiler and, code optimization techniques, which are independent of target machines. Likewise, the process of linking together independently assembled subprograms does not usually depend on the computer being used.

1.2 The Simplified Instructional Computer (SIC)

Simplified Instructional Computer (SIC) is a hypothetical computer that includes the hardware features most often found on real machines. There are two versions of SIC, they are, standard model (SIC), and, extension version (SIC/XE) (extra equipment or extra expensive).

1.2.1 SIC Machine Architecture

We discuss here the SIC machine architecture with respect to its Memory and Registers, Data Formats, Instruction Formats, Addressing Modes, Instruction Set, Input and Output

Memory

There are 2^{15} bytes in the computer memory, that is 32,768 bytes , It uses Little Endian format to store the numbers, 3 consecutive bytes form a word , each location in memory contains 8-bit bytes.

Registers

There are five registers, each 24 bits in length. Their mnemonic, number and use are given in the following table.

Mnemonic	Number	Use
A	0	Accumulator; used for arithmetic operations
X	1	Index register; used for addressing
L	2	Linkage register; JSUB
PC	8	Program counter
SW	9	Status word, including CC

Data Formats

Integers are stored as 24-bit binary numbers , 2's complement representation is used for negative values, characters are stored using their 8-bit ASCII codes, No floating-point hardware on the standard version of SIC.

Instruction Formats

Opcode(8)   x   Address (15)

All machine instructions on the standard version of SIC have the 24-bit format as shown above

Addressing Modes

Mode	**Indication**	**Target address calculation**
Direct	x = 0	TA = address
Indexed	x = 1	TA = address + (x)

There are two addressing modes available, which are as shown in the above table. Parentheses are used to indicate the contents of a register or a memory location.

Instruction Set

SIC provides, load and store instructions (LDA, LDX, STA, STX, etc.). Integer arithmetic operations: (ADD, SUB, MUL, DIV, etc.). All arithmetic operations involve register A and a word in memory, with the result being left in the register. Two instructions are provided for subroutine linkage. COMP compares the value in register A with a word in memory, this instruction sets a condition code CC to indicate the result. There are conditional jump instructions: (JLT, JEQ, JGT), these instructions test the setting of CC and jump accordingly. JSUB jumps to the subroutine placing the return address in register L, RSUB returns by jumping to the address contained in register L.

Input and Output

Input and Output are performed by transferring 1 byte at a time to or from the rightmost 8 bits of register A (accumulator). The Test Device (TD) instruction tests whether the addressed device is ready to send or receive a byte of data. Read Data (RD), Write Data (WD) are used for reading or writing the data.

Data movement and Storage Definition

LDA, STA, LDL, STL, LDX, STX (A- Accumulator, L – Linkage Register, X – Index Register), all uses 3-byte word. LDCH, STCH associated with characters uses 1-byte. There are no memory-memory move instructions.

Storage definitions are

- WORD - ONE-WORD CONSTANT
- RESW - ONE-WORD VARIABLE
- BYTE - ONE-BYTE CONSTANT
- RESB - ONE-BYTE VARIABLE

Example Programs (SIC)

Example 1(Simple data and character movement operation)

```
           LDA FIVE
           STA  ALPHA
           LDCH       CHARZ
           STCH       C1
       .
ALPHA      RESW       1
FIVE       WORD       5
CHARZ      BYTE  C'Z'
C1         RESB       1
```

Example 2(Arithmetic operations)

```
          LDA  ALPHA
          ADD  INCR
          SUB  ONE
          STA  BEETA
          ……..
          ……..
          ……..
          ……..
ONE       WORD   1
ALPHA      RESW   1
BEETA     RESW    1
INCR      RESW    1
```

Example 3(Looping and Indexing operation)

```
LDX      ZERO         :   X = 0
MOVECH   LDCH   STR1, X       :   LOAD A FROM STR1
STCH    STR2, X     :   STORE A TO STR2
TIX       ELEVEN    :    ADD 1 TO X, TEST
JLT       MOVECH
 .
```

```
                .
                .
STR1          BYTE     C ‘HELLO WORLD’
STR2          RESB     11
ZERO          WORD    0
ELEVEN        WORD   11
```

Example 4(Input and Output operation)

```
INLOOP     TD       INDEV            : TEST INPUT DEVICE
           JEQ     INLOOP          : LOOP UNTIL DEVICE IS READY
           RD      INDEV           : READ ONE BYTE INTO A
           STCH DATA               :  STORE A TO DATA
           .
           .
OUTLP      TD       OUTDEV          : TEST OUTPUT DEVICE
           JEQ      OUTLP           : LOOP UNTIL DEVICE IS READY
           LDCH   DATA             : LOAD DATA INTO A
           WD      OUTDEV         : WRITE A TO OUTPUT DEVICE
            .
            .
INDEV       BYTE      X ‘F5’          : INPUT DEVICE NUMBER
OUTDEV    BYTE      X ‘08’          : OUTPUT DEVICE NUMBER
DATA        RESB      1                : ONE-BYTE VARIABLE
```

Example 5 (To transfer two hundred bytes of data from input device to memory)

```
LDX    ZERO
CLOOP       TD       INDEV
            JEQ      CLOOP
            RD       INDEV
            STCH   RECORD, X
            TIX      B200
            JLT     CLOOP
            .
            .
INDEV       BYTE     X ‘F5’
RECORD    RESB     200
ZERO        WORD    0
B200         WORD    200
```

Example 6 (Subroutine to transfer two hundred bytes of data from input device to memory)

```
            JSUB  READ
            ………….
            ………….
READ        LDX    ZERO
CLOOP       TD     INDEV
            JEQ    CLOOP
            RD     INDEV
            STCH   RECORD, X
            TIX    B200          : add 1 to index compare 200 (B200)
            JLT    CLOOP
            RSUB
            ……..
            ……..
INDEV       BYTE    X ‘F5’
RECORD      RESB   200
ZERO        WORD    0
B200        WORD    200
```

1.2.2 SIC/XE Machine Architecture

Memory

Maximum memory available on a SIC/XE system is 1 Megabyte (2^{20} bytes)

Registers

Additional B, S, T, and F registers are provided by SIC/XE, in addition to the registers of SIC

Mnemonic	Number	Special use
B	3	Base register
S	4	General working register
T	5	General working register
F	6	Floating-point accumulator (48 bits)

Floating-point data type

There is a 48-bit floating-point data type, $F*2^{(e-1024)}$

1	11	36
s	exponent	fraction

Instruction Formats

The new set of instruction formats fro SIC/XE machine architecture are as follows. Format 1 (1 byte): contains only operation code (straight from table). Format 2 (2 bytes): first eight bits for operation code, next four for register 1 and following four for register 2. The numbers for the registers go according to the numbers indicated at the registers section (ie, register T is replaced by hex 5, F is replaced by hex 6). Format 3 (3 bytes): First 6 bits contain operation code, next 6 bits contain flags, last 12 bits contain displacement for the address of the operand. Operation code uses only 6 bits, thus the second hex digit will be affected by the values of the first two flags (n and i). The flags, in order, are: n, i, x, b, p, and e. Its functionality is explained in the next section. The last flag e indicates the instruction format (0 for 3 and 1 for 4). Format 4 (4 bytes): same as format 3 with an extra 2 hex digits (8 bits) for addresses that require more than 12 bits to be represented.

Format 1 (1 byte)

8
op

Format 2 (2 bytes)

8	**4**	**4**
op	**r1**	**r2**

Formats 1 and 2 are instructions do not reference memory at all

Format 3 (3 bytes)

6	**1**	**1**	**1**	**1**	**1**	**1**	**12**
op	**n**	**i**	**x**	**b**	**p**	**e**	**disp**

Format 4 (4 bytes)

6	1	1	1	1	1	1	20
op	**n**	**i**	**x**	**b**	**p**	**e**	**address**

Addressing modes & Flag Bits

Five possible addressing modes plus the combinations are as follows.

Direct (x, b, and p all set to 0): operand address goes as it is. n and i are both set to the same value, either 0 or 1. While in general that value is 1, if set to 0 for format 3 we can assume that the rest of the flags (x, b, p, and e) are used
as a part of the address of the operand, to make the format compatible to the
SIC format

Relative (either b or p equal to 1 and the other one to 0): the address of the operand should be added to the current value stored at the B register (if b = 1) or to the value stored at the PC register (if p = 1)

Immediate (i = 1, n = 0): The operand value is already enclosed on the instruction (ie. lies on the last 12/20 bits of the instruction)

Indirect (i = 0, n = 1): The operand value points to an address that holds the address for the operand value.

Indexed (x = 1): value to be added to the value stored at the register x to obtain real address of the operand. This can be combined with any of the previous modes except immediate.

The various flag bits used in the above formats have the following meanings

e - e = 0 means format 3, e = 1 means format 4

Bits x,b,p: Used to calculate the target address using relative, direct, and indexed addressing Modes

Bits i and n: Says, how to use the target address

b and p - both set to 0, disp field from format 3 instruction is taken to be the target address. For a format 4 bits b and p are normally set to 0, 20 bit address is the target address

x - x is set to 1, X register value is added for target address calculation

i=1, n=0 Immediate addressing, **TA**: TA is used as the operand value, no memory reference
i=0, n=1 Indirect addressing, **((TA))**: The word at the TA is fetched. Value of TA is taken as the address of the operand value

i=0, n=0 or i=1, n=1 Simple addressing, **(TA)**:TA is taken as the address of the operand value

Two new relative addressing modes are available for use with instructions assembled using format 3.

Mode	**Indication**	**Target address calculation**
Base relative	b=1,p=0	TA=(B)+ disp (0≤disp ≤4095)
Program-counter relative	b=0,p=1	TA=(PC)+ disp (-2048≤disp ≤2047)

Instruction Set

SIC/XE provides all of the instructions that are available on the standard version. In addition we have, Instructions to load and store the new registers LDB, STB, etc, Floating-point arithmetic operations, ADDF, SUBF, MULF, DIVF, Register move instruction : RMO, Register-to-register arithmetic operations, ADDR, SUBR, MULR, DIVR and, Supervisor call instruction : SVC.

Input and Output

There are I/O channels that can be used to perform input and output while the CPU is executing other instructions. Allows overlap of computing and I/O, resulting in more efficient system operation. The instructions SIO, TIO, and HIO are used to start, test and halt the operation of I/O channels.

Example Programs (SIC/XE)

Example 1 (Simple data and character movement operation)

```
                LDA #5
                STA  ALPHA
                LDA       #90
                STCH          C1
             .
             .
ALPHA           RESW         1
C1            RESB  1
```

Example 2(Arithmetic operations)

```
        LDS   INCR
        LDA  ALPHA
        ADD  S,A
        SUB  #1
        STA  BEETA
        ………….
       …………..
ALPHA RESW   1
BEETA RESW   1
INCR    RESW   1
```

Example 3(Looping and Indexing operation)

```
                 LDT      #11
                 LDX      #0                :   X = 0
 MOVECH      LDCH   STR1, X       :   LOAD A FROM STR1
                 STCH    STR2, X      :   STORE A TO STR2
                TIXR      T                :    ADD 1 TO X, TEST (T)
                JLT        MOVECH
                ……….
                ……….
                ………
    STR1          BYTE      C ‘HELLO WORLD’
    STR2          RESB      11
```

Example 4 (To transfer two hundred bytes of data from input device to memory)

```
             LDT     #200
             LDX    #0
CLOOP      TD       INDEV
             JEQ     CLOOP
             RD       INDEV
             STCH   RECORD, X
             TIXR     T
             JLT      CLOOP
             .
             .
INDEV        BYTE      X ‘F5’
RECORD    RESB     200
```

Example 5 (Subroutine to transfer two hundred bytes of data from input device to memory)

```
            JSUB   READ
             ……….
             ……….
READ         LDT    #200
            LDX    #0
CLOOP        TD      INDEV
             JEQ    CLOOP
             RD      INDEV
             STCH   RECORD, X
             TIXR    T                : add 1 to index compare T
             JLT     CLOOP
             RSUB
             ……..
             ……..
INDEV        BYTE     X ‘F5’
RECORD       RESB    200
```

1.3 Different Architectures

The following section introduces the architectures of CISC and RISC machines. CISC machines are called traditional machines. In addition to these we have recent RISC machines. Different machines belonging to both of these architectures are compared with respect to their Memory, Registers, Data Formats, Instruction Formats, Addressing Modes, Instruction Set, Input and Output

1.3.1 CISC machines

Traditional (CISC) Machines, are nothing but, Complex Instruction Set Computers, has relatively large and complex instruction set, different instruction formats, different lengths, different addressing modes, and implementation of hardware for these computers is complex. VAX and Intel x86 processors are examples for this type of architecture.

1.3.1.1 VAX Architecture

Memory - The VAX memory consists of 8-bit bytes. All addresses used are byte addresses. Two consecutive bytes form a word, Four bytes form a longword, eight bytes form a quadword, sixteen bytes form a octaword. All VAX programs operate in a virtual address space of 232 bytes , One half is called system space, other half process space.

Registers – There are 16 general purpose registers (GPRs) , 32 bits each, named as R0 to R15, PC (R15), SP (R14), Frame Pointer FP (R13), Argument Pointer AP (R12) ,Others available for general use. There is a Process status longword (PSL) – for flags.

Data Formats - Integers are stored as binary numbers in byte, word, longword, quadword, octaword. 2's complement notation is used for storing negative numbers. Characters are stored as 8-bit ASCII codes. Four different floating-point data formats are also available.

Instruction Formats - VAX architecture uses variable-length instruction formats – op code 1 or 2 bytes, maximum of 6 operand specifiers depending on type of instruction. Tabak – Advanced Microprocessors (2nd edition) McGraw-Hill, 1995, gives more information.

Addressing Modes - VAX provides a large number of addressing modes. They are Register mode, register deferred mode, autoincrement, autodecrement, base relative, program-counter relative, indexed, indirect, and immediate.

Instruction Set – Instructions are symmetric with respect to data type - Uses prefix – type of operation, suffix – type of operands, a modifier – number of operands. For example, ADDW2 - add, word length, 2 operands, MULL3 - multiply, longwords, 3 operands CVTCL - conversion from word to longword. VAX also provides instructions to load and store multiple registers.

Input and Output - Uses I/O device controllers. Device control registers are mapped to separate I/O space. Software routines and memory management routines are used for input/output operations.

1.3.1.2 Pentium Pro Architecture

Introduced by Intel in 1995.

Memory - consists of 8-bit bytes, all addresses used are byte addresses. Two consecutive bytes form a word, four bytes form a double word (dword). Viewed as collection of segments, and, address = segment number + offset. There are code, data, stack , extra segments.

Registers – There are 32-bit, eight GPRs, namely EAX, EBX, ECX, EDX, ESI, EDI, EBP, ESP. EAX, EBX, ECX, EDX – are used for data manipulation, other four are used to hold addresses. EIP – 32-bit contains pointer to next instruction to be executed. FLAGS is an 32 - bit flag register. CS, SS, DS, ES, FS, GS are the six 16-bit segment registers.

Data Formats - Integers are stored as 8, 16, or 32 bit binary numbers, 2's complement for negative numbers, BCD is also used in the form of unpacked BCD, packed BCD. There are three floating point data formats, they are single, double, and extended-precision. Characters are stored as one per byte – ASCII codes.

Instruction Formats – Instructions uses prefixes to specify repetition count, segment register, following prefix (if present), an opcode (1 or 2 bytes), then number of bytes to specify operands, addressing modes. Instruction formats varies in length from 1 byte to 10 bytes or more. Opcode is always present in every instruction

Addressing Modes - A large number of addressing modes are available. They are immediate mode, register mode, direct mode, and relative mode. Use of base register, index register with displacement is also possible.

Instruction Set – This architecture has a large and complex instruction set, approximately 400 different machine instructions. Each instruction may have one, two or three operands. For example Register-to-register, register-to-memory, memory-to-memory, string manipulation, etc…are the some the instructions.

Input and Output - Input is from an I/O port into register EAX. Output is from EAX to an I/O port

1.3.2 RISC Machines

RISC means Reduced Instruction Set Computers. These machines are intended to simplify the design of processors. They have Greater reliability, faster execution and less expensive processors. And also they have standard and fixed instruction length. Number of machine instructions, instruction formats, and addressing modes relatively small. UltraSPARC Architecture and Cray T3E Architecture are examples of RISC machines.

1.3.2.1 UltraSPARC Architecture

Introduced by Sun Microsystems. SPARC – Scalable Processor ARChitecture. SPARC, SuperSPARC, UltraSPARC are upward compatible machines and share the same basic structure.

Memory - Consists of 8-bit bytes, all addresses used are byte addresses. Two consecutive bytes form a halfword, four bytes form a word , eight bytes form a double word. Uses virtual address space of 2^{64} bytes, divided into pages.

Registers - More than 100 GPRs, with 64 bits length each called Register file. There are 64 double precision floating-point registers, in a special floating-point unit (FPU). In addition to these, it contains PC, condition code registers, and control registers.

Data Formats - Integers are stored as 8, 16, 32 or 64 bit binary numbers. Signed, unsigned for integers and 2’s complement for negative numbers. Supports both big-endian and little-endian byte orderings. Floating-point data formats – single, double and quad-precision are available. Characters are stored as 8-bit ASCII value.

Instruction Formats - 32-bits long, three basic instruction formats, first two bits identify the format. Format 1 used for call instruction. Format 2 used for branch instructions. Format 3 used for load, store and for arithmetic operations.
Addressing Modes - This architecture supports immediate mode, register-direct mode,PC-relative, Register indirect with displacement, and Register indirect indexed.

Instruction Set – It has fewer than 100 machine instructions. The only instructions that access memory are loads and stores. All other instructions are register-to-register operations. Instruction execution is pipelined – this results in faster execution, and hence speed increases.

Input and Output - Communication through I/O devices is accomplished through memory. A range of memory locations is logically replaced by device registers. When a load or store instruction refers to this device register area of memory, the corresponding device is activated. There are no special I/O instructions.

1.3.2.2 Cray T3E Architecture

Announced by Cray Research Inc., at the end of 1995 and is a massively parallel processing (MPP) system, contains a large number of processing elements (PEs), arranged in a three-dimensional network. Each PE consists of a DEC Alpha EV5 RISC processor, and local memory.

Memory - Each PE in T3E has its own local memory with a capacity of from 64 megabytes to 2 gigabytes, consists of 8-bit bytes, all addresses used are byte addresses. Two consecutive bytes form a word, four bytes form a longword, eight bytes form a quadword.

Registers – There are 32 general purpose registers(GPRs), with 64 bits length each called R0 through R31, contains value zero always. In addition to these, it has 32 floating-point registers, 64 bits long, and 64-bit PC, status , and control registers.

Data Formats - Integers are stored as long and quadword binary numbers. 2's complement notation for negative numbers. Supports only little-endian byte orderings. Two different floating-point data formats – VAX and IEEE standard. Characters stored as 8-bit ASCII value.

Instruction Formats - 32-bits long, five basic instruction formats. First six bits always identify the opcode.

Addressing Modes - This architecture supports, immediate mode, register-direct mode, PC-relative, and Register indirect with displacement.

Instruction Set - Has approximately 130 machine instructions. There are no byte or word load and store instructions. Smith and Weiss – "PowerPC 601 and Alpha 21064: A Tale of TWO RISCs " – Gives more information.

Input and Output - Communication through I/O devices is accomplished through multiple ports and I/O channels. Channels are integrated into the network that interconnects the processing elements. All channels are accessible and controllable from all PEs.