CS202 – System Software

Dr. Manish Khare

Lecture 2

SIC Machine Architecture

- Memory
  - 2^{15} bytes in the computer memory
  - 3 consecutive bytes form a work
  - 8-bit bytes

- Registers

Mnemonic	Number	Special
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 accumulated as 24-bit binary numbers; 2's complement demonstration is used for negative values
- No floating-point hardware.

Instruction Formats

Opcode(8)	x	Address(15)

Addressing Modes

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

Instruction Set

- integer arithmetic operations: ADD, SUB, MUL, DIV, etc.
  - All arithmetic operations entail register A and a word in memory, with the consequence being left in the register.
- comparison: COMP
  - COMP compares the value in register A with a word in memory, this instruction sets a condition code CC to specify the result
- conditional jump instructions: JLT, JEQ, JGT
  - these instructions test the setting of CC and jump accordingly
- subroutine linkage: JSUB, RSUB
  - JSUB jumps to the subroutine, positioning the return address in register L
  - RSUB returns by jumping to the address enclosed in register L

Input and Output

- Input and output are executed by conveying 1 byte at a time to or from the rightmost 8bits of register A
- The Test Device (TD) instruction tests whether the addressed device is prepared to send or obtain a byte of data
- Read Data (RD)
- Write Data (WD)

Contents of status word register

Bit position	Field name	Use
0	MODE	0=user mode, 1=supervisor mode
1	IDLE	0=running, 1=idle
2~5	ID	Process identifier
6~7	CC	Condition code
8~11	MASK	Interrupt mask
12~15		Unused
16~23	ICODE	Interruption code

SIC/XE Machine Architecture

Memory

- Almost the similar as formerly described for SIC
- However 1 MB(220 bytes)

More Registers

Mnemonic	Number	Special use
B	3	Base register; used for addressing
S	4	General working register
T	5	General working register
F	6	Floating-point acumulator (48bits)

- Data Formats:
  - Same data format
  - In addition 48-bit floating point data type:
  - $frac*2^{(exp-1024)}$
  - Frac: 0~1
  - Exp: 0~2047

Instruction Formats

Format 1		
op(8)		
Format 2		
op(8)	**r1(4)**	**r2(4)**

Format 3						e=0	
op(6)	n	I	x	b	p	e	disp(12)

Format 4						e =1	
op(6)	n	I	x	b	p	e	Address(20)

Instruction Set

- new registers: LDB, STB, etc.
- floating-point arithmetic: ADDF, SUBF, MULF, DIVF
- register move: RMO
- register-register arithmetic: ADDR, SUBR, MULR, DIVR
- supervisor call: SVC
- produces an interrupt for OS

Input/Output

- SIO, TIO, HIO: start, test, halt the operation of I/O device

SIC/XE Instruction formats

- Bigger memory means an address cannot fit into a 15-bit field

  Expand addressing capacity
- Use some form of relative addressing-> instruction format 3
- Enlarge the address field to 20 bits-> instruction format 4

  Additional instructions do not reference memory
- Instruction format 1 & 2

Traditional (CISC) Machine

- CISC Machines generally have a relatively large and complicated instruction set, several different instruction formats and length, and many different addressing mode.
- The implementation of such an architecture in hardware tends to be complex.

VAX Architecture

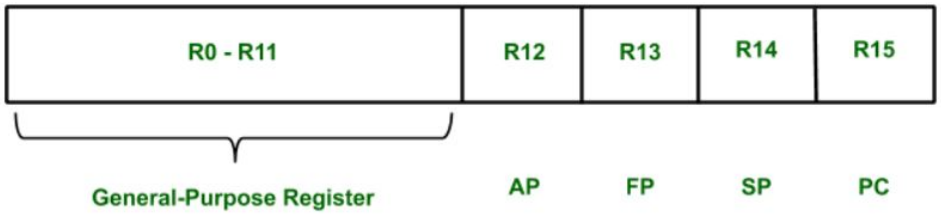
- **VAX (Virtual Address eXtension) Architecture** was designed to increase the compatibility by improving the hardware of the earlier designed machines. As VAX architecture is an example of the CISC (Complex Instruction Set Computers) therefore there are large and complicated instruction sets used in the system.

Memory

- The VAX memory consists of 8-bit bytes.
- All addresses used are byte addresses.
- Two consecutive bytes form a word; four byte form a longword; eight bytes form a quadword; sixteen bytes form an octaword.
- All VAX programs operates on Virtual Address Space (2^{32} bytes). The Virtual Address Space is divided into two spaces:
  - System Space: which contains the operating system and is shared by all programs
  - Process Space: is defined separately for each program

Registers

- VAX architecture have 16 general-purpose registers from R0 to R15. Some of these registers have special name and use.



AP - *Argument Pointer*
FP - *Frame Pointer*
SP - *Stack Pointer*
PC - *Program Counter*

Data Formats

- Integers are stored as Binary numbers in byte, word, longword, quadword or octaword.
- Characters are represented using 8-bit ASCII codes.
- Floating points are represented using four different floating-point formats of length ranging from 4 to 16 bytes.

Instruction Formats

- VAX machine architecture use a variable-length instruction format. Each instruction consists of an operand code (1 or 2 bytes) followed by up to six operand specifier, depending on the type of instruction.

Addressing Modes:

- VAX architecture use a large number of addressing modes.
- There are number of modes available such as register mode, register deferred mode, autoincrement and autodecrement mode.
- There are also base relative addressing modes, with displacement fields of different lengths. Program counter relative mode is also used to deal with PC register.

Instruction Set:

- In VAX systems instruction mnemonics are formed by combining following elements:
  - **Prefix:** A Prefix specifies the type of operation.
  - **Suffix:** A Suffix specifies the data type of the operands.
  - **Modifier:** A modifier specifies the number of operand involved.

Input and Output

- I/O device controller are used to implement I/O on VAX architecture. Each controller has a set of control/status. The portion of the space into which the device controller register are mapped is called I/O space.

Pentium Pro Architecture

- The **Pentium Pro microprocessor** belongs to the CISC (Complex Instruction Set Computers) machines.
- Processor of Pentium Pro family are mostly present in majority of personal computers.
- The term 'Pentium processor' refers to an Intel x86 family of microprocessors that share a common architecture and instruction set.

Silent features of Pentium Pro Architecture:

- 64 bit data bus
- 8 bytes of data information can be transferred to and from memory in a single bus cycle
- Supports burst read and burst write back cycles
- Supports pipelining
- Instruction cache

Memory:

- Pentium Pro microprocessor have 8-bit byte memory. Two consecutive bytes form a word, four consecutive bytes form a double word. All the address in the memory are of byte formats.
- Programmers usually view Pentium Pro (x86) memory as a collection of segments.
- Address consists two parts: Segment number and an offset that points to a byte within the segment.
  - Segments can be of different sizes, and are often used for different purposes.

Registers

- There are 8 general-purpose registers present in the Pentium Pro architecture. Each register is 32-bit long. First four register are used for data manipulation and next four register are used to hold address. There are some special-purpose registers in the x86 architecture such as Segment register, FLAGS register and EIP register.

Data Formats

- Integers are stored as 8-, 16-, 32-bit Binary numbers.
- Characters are represented using 8-bit ASCII codes.
- Floating points are represented using three different formats namely single-precision format, double-precision format, extended-precision format.

Instruction Formats:

- All the Pentium Pro instruction uses basic format.
- The basic format for Pentium Pro Instructions is:
  - **Prefix:** A Prefix specifies the operation of an instruction.
  - **Suffix:** A Suffix specifies the data type of the operands.
  - **Modifier:** A modifier specifies the number of operand involved.

- The Opcode is the only element that is present in every instruction. Other elements may or may not present or may be of different lengths depending on the operations of instructions.

Addressing Modes

- Pentium Pro Architecture have a very large number of Addressing modes.
- Operands value is specified either by using Immediate mode or by using register mode.
- Operands stored in memory are specified using variation of the Target address (TA) calculation:

```
TA = (base register) + (index register) * (scale factor) + displacement
```

- **Base register:** Any general-purpose register may be used as a base register.
- **Index register:** Any general-purpose register except ESP can be used as an index register.
- **Scale Factor:** Scale factor may have values 1, 2, 4 or 8.
- **Displacement:** Displacement may have 8-, 18-, or 32-bit values.

Instruction Set:

- Pentium Pro Architecture has a large and complex instruction set having more than 400 different machine instructions.
- An instruction may have zero, one, two, or three operands.
- There are Register-to-Register instructions, Register-to-Memory instructions and Memory-to-Memory instructions.
- In Pentium Pro Architecture there are some special-purpose registers to perform operations required in the high-level programming languages.

Input and Output

- Input instruction transfer one byte, word or doubleword at a time from an I/O port into register EAX. Output instruction transfer one byte, word or doubleword from EAX to an I/O port. Entire string can be transferred using single operation.

RISC Machines

- The main idea behind this is to make hardware simpler by using an instruction set composed of a few basic steps for loading, evaluating, and storing operations just like a load command will load data, a store command will store the data.
- Reduce the cycles per instruction at the cost of the number of instructions per program.
  - UltraSPARC Architecture
  - PowerPC Architecture

UltraSPARC Architecture

- **UltraSPARC Architecture** belongs to the SPARC (Scalable Processor Architecture) family of processors.
- This architecture is suitable for wide range of microcomputers and supercomputers.
- UltraSPARC is example of RISC (Reduced Instruction Set Computer).

Memory:

- Memory consists of 8 bit-bytes. Two consecutive bytes form a halfword, four bytes form a word, eight bytes form a doubleword.
- UltraSPARC programs operates on **Virtual Address Space** (2^{64} bytes).
- Virtual Address Space is divided into pages and these pages are stored in the physical memory or on disk.

Registers:

- UltraSPARC architecture include a large file of registers that have more than 100 general purpose registers.
- Any procedure can access only 32 registers only.
- The SPARC hardware uses window into registers file to manage all the operations of different procedures.
- Beside these register files, UltraSPARC also uses Program Counter, code register, and other control registers.

Data Formats:

- Integers are stored as 8-, 16-, 32-, or 64-bit Binary numbers.
- Characters are represented using 8-bit ASCII codes.
- Floating points are represented using three different formats namely single-precision format, double-precision format, quad-precision format.

Instruction Formats:

- SPARC architecture use three basic instruction formats.
- All the instructions are of 32-bit long and first two bits are used to identify which format is being used.

Format 1		
op(8)		
Format 2		
op(8)	**r1(4)**	**r2(4)**

Format 3- Used by all the remaining instructions like register load and store.


```
n=Indirect mode,
i=Immediate addressing,
x=Index addressing,
b=Base addressing,
p= Program counter,
e=Exponential addressing
```

Addressing Modes:

- Operands in memory are addressed using one of the following three modes:

Mode	Target address(TA) calculation
PC-relative	TA=(PC) + displacement
Register indirect with displacement	TA=(register) + displacement
Register indirect indexed	TA=(register-1) + (register-2)

PC-relative is used only for branch instructions

Instruction set:

- This architecture have less number of instructions as compared to CISC machines.
- The only instructions that access memory are load and stores.
- All other instructions operates on register only. Instruction execution on a SPARC system is pipelined which means while one instruction is executed next one is being fetched from memory and decoded

Input and Output:

- Communication between I/O devices and SPARC operation are accomplished through memory.
- Input and Output can be performed with the regular instruction set of the computer, and no special I/O instructions are needed

PowerPC Architecture

- **PowerPC Architecture** are microprocessor for personal computers. PowerPC is a RISC (Reduced Instruction Set Computer) architecture which are very powerful and low-cost microprocessors. RISC architecture tries to keep the processor as busy as possible.
- Design features of PowerPC are as follows:
  - Broad range implementation
  - Simple processor design
  - Superscalar architecture
  - Multiprocessor features
  - 64-bit architecture
  - Support for operation in both big-endian and little-endian mode. PowerPC can switch from one mode to another at run time.
  - Separate set of Floating Point Registers (FPRs) for floating-point instructions

Memory:

- Memory consists of 8-bit bytes.
- Two consecutive bytes form a halfword, four bytes form a word, eight bytes form a doubleword, sixteen bytes form a quadword.
- PowerPC programs can be written using a **Virtual Address Space** (2^{64} bytes).
- Address space are divided into fixed-length segments which are further divided into pages.

- Registers:
  - There are 32 general-purpose registers (GPR) from GPR0 to GPR31. Length of each register is 64-bit.
  - The general purpose register are used to store and manipulate data and addresses.
  - As PowerPC machine support floating point data format so it have Floating-point unit (FPU) for computation.
  - Some of the register's supported by PowerPC architecture are:

Register	**Operations**
Link Register(LR)	Contain address to return at the end of the function call
Condition Register (CR)	Signify the result of an instruction
Count Register (CTR)	For Loop count

Data Formats:

- Integers are stored as 8-, 16-, 32-, or 64-bit Binary numbers.
- Characters are represented using 8-bit ASCII codes.
- Floating points are represented using two different floating-point formats namely single-precision format and double-precision format.

Instruction Formats:

- PowerPC support seven basic instruction formats.
- All of these instruction formats are 32-bits long.
- PowerPC architecture instruction format have more variety and complexity as compared to other RISC systems such as SPARC.
- Bit numbering for PowerPC is the opposite of most other definitions:

```
bit 0 is the most significant bit,
and bit 31 is the least significant bit
```

- Instructions are first decoded by the upper 6 bits in a field, called the primary opcode. The remaining 26 bits contain fields for operand specifiers, immediate operands, and extended opcodes, and these may be reserved bits or fields.

Addressing Mode:

- Load and store operations use one of the following three addressing mode depending upon the operand value:

Mode	Target address(TA) calculation
Register indirect	TA=(register)
Register indirect with index	TA=(register-1) + (register-2)
Register indirect with immediate index	TA=(register) + displacement

- Branch instructions use one of the following three addressing modes:

Mode	**Target address(TA) calculation**
Absolute	TA=actual address
Relative	TA=current instruction address + displacement
Link Register	TA=(LR)
Count Register	TA=(CR)

Instruction Set

- PowerPC architecture is more complex than the other RISC systems.
- Thus PowerPC architecture has approximately 200 machine instructions.
- This architecture follows the pipeline execution of instructions which means while one instruction is executed next one is being fetched from memory and decoded.

Input and Output:

- PowerPC architecture follows two different methods for performing I/O operations.
- In one approach Virtual address space is used while in the other approach I/O is performed using Virtual memory management.

Major Components of a Programming System

- The Major Components of a programming system is:
  - Operating system
  - Language translators
    - Compilers
    - Interpreters
    - Assemblers
    - Preprocessors
    - Loaders
    - Linkers
    - Macro processors.