Presentation on
SUN SPARC Microprocessor

Department:
BE 5th Sem Computer

Subject:
Microprocessor and Interfacing
(2150707)
CONTENT

- INTRODUCTION
- DESIGN GOALS
- HISTORY
- THE SPARC ARCHITECTURE
  - Integer Unit (IU)
  - The Register Window
  - Floating Point Unit (FPU)
  - Coprocessor
- INSTRUCTIONS
- SPARC V9 FEATURES
Introduction

- **SPARC** stands for Scalable Processor Architecture.
- developed by Sun Microsystems in the 1980s.
- is based on the RISC structure designed at the University of California at Berkeley in early 1980s.
- The SPARC architecture is a non-proprietary architecture that any person or company can license and use to develop microprocessors and other semiconductor devices based on published industry standards.
- In 1989, Sun Microsystems transferred ownership of the **SPARC** specifications to an independent, non-profit organization, **SPARC International**, which administers and licenses the technology and provides conformance testing and other services for its members.
Design Goals

- SPARC was designed as a target for optimizing compilers and easily pipelined hardware implementations. SPARC implementations provide exceptionally high execution rates (MIPS) and short time-to-market development schedules.

- Provide the scalability of the cost/performance ratio of successive implementations with the current improvements in circuit technology.

- The "Scalable" in SPARC comes from the fact that the SPARC specification allows implementations to scale from processors required in embedded systems to processors used for servers.
Brief History

- 3 major revisions to the SPARC architecture
  - SPARC-V7, 32bit, 1986
  - SPARC-V8, 32bit, 1990
  - SPARC-V9, 64bit, 1993

- Backward binary compatibility between them.

The SPARC Architecture

- It is a Load and store architecture. Operations are always done over registers.
- Uses “register window” concept thus offering a large number of registers.
- Uses delay slot to optimize branch instruction.
- Passes arguments using registers and the stack.
Fig. Architecture of sun sparc
The Integer Unit (IU)

- Contains the general purpose registers and controls the overall operation of the processor.

- May contain from 64 to 528 general-purpose 64-bit registers. They are partitioned into 8 global registers, 8 alternate global registers, plus a circular stack of from 3 to 32 sets of 16 registers each, known as register windows.

- Executes the integer arithmetic instructions and computes memory addresses for loads and stores.

- Maintains the program counters and controls instruction execution for the FPU.
The Register Window

- At any time, an instruction can access the 8 global registers and a 24-register window.

- A register window comprises a 16-register set divided into 8 in and 8 local registers together with the 8 in registers of an adjacent register set, addressable from the current window as its out registers.

- When a procedure is called, the register window shifts by sixteen registers, hiding the old input registers and old local registers and making the old output registers the new input registers.

- **Input registers**: arguments are passed to a function.

- **Local registers**: to store any local data.

- **Output registers**: When calling a function, the programmer puts his argument in these registers.
The Register Window

- The current window into the *r* registers is given by the current window pointer (CWP) register.
The Floating-point Unit (FPU)

- The FPU has 32 32-bit (single-precision) floating-point registers, 32 64-bit (double-precision) floating-point registers, and 16 128-bit (quad-precision) floating-point registers.

- Double-precision values occupy an even-odd pair of single-precision registers.

- Quad-precision values occupy an odd-even number pair of double precision registers.

- Floating-point load/store instructions are used to move data between the FPU and memory.

- The memory address is calculated by the IU.

- Floating-Point operate (FPop) instructions perform the floating-point arithmetic operations and comparisons.
The instruction set includes support for a single, implementation-dependent coprocessor. The coprocessor has its own set of registers.

Coprocessor load/store instructions are used to move data between the coprocessor registers and memory.

Floating-point instructions mirrors coprocessor instructions.

Not implemented in SPARC V9.
Instructions

Instructions can fall into following basic categories:

- Load/store
- Arithmetic/logical/shift
- Control transfer
- Read/write control register
- Floating-point/Coprocessor operate
SPARC v9 features

- 64-bit Data and Addresses as compared to 32-bit Data and Addresses of SPARC V8.
- 32 double-precision floating-point registers,
- Software-settable branch prediction
- 64-bit integer multiply and divide instructions
- Load/store floating-point quad word instructions
- Branches on register value (eliminating the need to compare)
- The V9 remains binary compatible with all previous SPARC architecture.
Thank You...
Vidhyadeep Institute of Engineering & Technology

**Branch**: Computer  
**Year**: $3^{rd}$ ($5^{th}$ sem)  
**Subject**: MPI
Topic:-

8255A Microprocessor
Features of 8255A:-

- The 8255A is a widely used, programmable, peripheral I/O device.
- It is compatible with all Intel and most other microprocessors.
- The 8255 can operate in 3 I/O modes:
  
  1. Mode 0: Simple Input/Output
  2. Mode 1: Input/Output With handshake
  3. Mode 2: Bi-directional I/O data Transfer
- **Mode 0:**

  In this mode, Port A and B is used as two 8-bit ports and Port C as two 4-bit ports. Each port can be programmed in either input mode or output mode where outputs are latched and inputs are not latched. Ports do not have interrupt capability.

- **Mode 1:**

  In this mode, Port A and B is used as 8-bit I/O ports. They can be configured as either input or output ports. Each port uses three lines from port C as handshake signals. Inputs and outputs are latched.

- **Mode 2:**

  In this mode, Port A can be configured as the bidirectional port and Port B either in Mode 0 or Mode 1. Port A uses five signals from Port C as handshake signals for data transfer. The remaining three signals from Port C can be used either as simple I/O or as handshake for port B.
Data Bus Buffer:-

This three-state bi-directional 8-bit buffer is used to interface the 8255 to the system data bus. Data is transmitted or received by the buffer upon execution of input or output instructions by the CPU.

Control Logic:-

The control logic block accepts control bus signals as well as inputs from the address bus, and issues commands to the individual group control blocks.

Group A and Group B Controls:-

Each of the Group A and Group B control blocks receives control words from the CPU and issues appropriate commands to the ports associated with it.
➢ Port A:

This has an 8-bit latched and buffered output and an 8-bit input latch. It can be programmed in three modes: mode 0, mode 1, mode 2.

➢ Port B:

This has an 8-bit data I/O latch/buffer and an 8-bit data input buffer. It can be programmed in mode 0 and mode 1.

➢ Port C:

This has one 8-bit unlatched input buffer and an 8-bit output latch/buffer. Port C can be separated into two parts and each can be used as control signals for ports A and B in the handshak mode.
Interfacing 8255 in i/o mapped i/o

Fig shows the interfacing of 8255 with 8085 in i/o mapped i/o technique.

Here read and write signals are activated when \( \overline{IOM} \) signal is high, indicating i/o

Bus cycle reset out signal from 8085 is connected to the RESET signal of the 8255.
Interfacing 8255 in memory mapped i/o

Fig shows the interfacing of 8255 with 8085 in memory mapped i/o technique. Here read and write signals are activated when io/m signal is low, indicating memory Bus cycle. To get absolute address, all remaining address lines are used to decode the Address for 8255. Other signal connections are same as in i/o mapped i/o.
Thank You
Subject:- Microprocessor And Interfacing
[2150707]

Topic:- 80286 Microprocessor
Internal Architecture
80286 Interface
80286 Microprocessor

- Announced in 1982, the 5th of i86 Family
- 125k transistors, HMOS III technology
- Two mode of operations
  - Real mode – operates as fast 8086/8088
  - Protected mode – enhances memory management,
- Multitasking and protection
- Improves both hardware and software
- Additional pipeline, demultiplexed address and data bus
- New enhanced instruction set (upward compat.)
- Pins are compatible with maximum mode of 8086
Internal Architecture

- **Super class**, **base class**, **parent class**: Terms to describe the parent in the relationship, which shares its functionality.
- **Subclass**, **derived class**, **child class**: Terms to describe the child in the relationship, which accepts functionality from its parent.
- **Extend**, **inherit**, **derive**: Become a subclass of another class.

- 4 independent units (8086 has only two units)
- 24-bit Address bus
- Up to 7 times higher performance than 8086
The 80286 consists of four separate processing units, They are:

- The Bus Unit (BU)
- The Address Unit (AU)
- The Execution Unit (EU)
- The Instruction Unit (IU)
The Bus Unit:

- The functions of the BU are:
  - To perform all memory and I/O read and writes.
  - To prefetch the instruction bytes.
  - To control the transfer of data to and from processor extension devices like the 80287 math co-processor.
The Address Unit:-

• It consists of segment register, an offset address and a physical address adder.
• The functions of the AU is to compute the physical address that will be sent out to the memory or I/O by the bus unit.
• We know the 80826 can operate in two different modes the real address mode and the protected virtual address mode.
The Execution Unit:-

• The execution unit includes the ALU, registers and the CPU, the registers consists of the general purpose register, index register pointers, flag registers and the 16-bit machine status word register (MSW).

• The functions of the EU are
  - to sequentially execute the instructions received from the instruction unit.
  - to direct the BU to access memory or I/O operands as needed.
The Instruction Unit:-

• The instruction unit includes the 3 instruction decoded queue and the instruction decoder.
• The function of the IU are:
  - to decode the prefetched instructions and hold them in an instruction queue, so that the EU can access them.
  - the IU decodes up to three(3) prefetched instructions. This helps the processor to speed up, as pipelining of instruction is done.
Internal Architecture

• Bus Unit generates all data, address and I/O signals.
• Prefetcher flushes the prefetched data, if IU finds a branch instruction.
• Address Unit (AU) off-loads address generation, translation and checking from BU.
• Instruction Unit off-loads EU by performing the instruction decoding.
• Execution Unit get the commands form IU, execute the instruction that maybe involve Registers, ALU, AU such as ‘ADD AX, [SI].’
80286 Interface
<table>
<thead>
<tr>
<th>Mnemonic</th>
<th>Function</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLK</td>
<td>System clock</td>
<td>I</td>
</tr>
<tr>
<td>D&lt;sub&gt;15&lt;/sub&gt; - D&lt;sub&gt;0&lt;/sub&gt;</td>
<td>Data bus</td>
<td>I/O</td>
</tr>
<tr>
<td>A&lt;sub&gt;23&lt;/sub&gt; - A&lt;sub&gt;0&lt;/sub&gt;</td>
<td>Address bus</td>
<td>O</td>
</tr>
<tr>
<td>BHE</td>
<td>Bus high enable</td>
<td>O</td>
</tr>
<tr>
<td>S&lt;sub&gt;1&lt;/sub&gt;, S&lt;sub&gt;0&lt;/sub&gt;</td>
<td>Bus cycle status</td>
<td>O</td>
</tr>
<tr>
<td>M/IO</td>
<td>Memory I/O select</td>
<td>O</td>
</tr>
<tr>
<td>COD/INTA</td>
<td>Code/interrupt acknowledge</td>
<td>O</td>
</tr>
<tr>
<td>LOCK</td>
<td>Bus lock</td>
<td>O</td>
</tr>
<tr>
<td>READY</td>
<td>Bus ready</td>
<td>I</td>
</tr>
<tr>
<td>HOLD</td>
<td>Bus hold</td>
<td>I</td>
</tr>
<tr>
<td>HLDA</td>
<td>Hold acknowledge</td>
<td>O</td>
</tr>
<tr>
<td>INTR</td>
<td>Interrupt request</td>
<td>I</td>
</tr>
<tr>
<td>NMI</td>
<td>Nonmaskable interrupt request</td>
<td>I</td>
</tr>
<tr>
<td>PEREQ</td>
<td>Processor extension request</td>
<td>I</td>
</tr>
<tr>
<td>PEACK</td>
<td>Processor extension acknowledge</td>
<td>O</td>
</tr>
<tr>
<td>BUSY</td>
<td>Processor extension busy</td>
<td>I</td>
</tr>
<tr>
<td>ERROR</td>
<td>Processor extension error</td>
<td>I</td>
</tr>
<tr>
<td>RESET</td>
<td>System reset</td>
<td>I</td>
</tr>
<tr>
<td>V&lt;sub&gt;SS&lt;/sub&gt;</td>
<td>System ground</td>
<td>I</td>
</tr>
<tr>
<td>V&lt;sub&gt;CC&lt;/sub&gt;</td>
<td>System power</td>
<td>I</td>
</tr>
</tbody>
</table>
Signal description of 80286

- It has 4 part:
  1) Memory I/O interface signals.
  2) Interrupt interface signals.
  3) DMA interface signals.
  4) Coprocessor interface signals.
Thank you