Short Notes Chapter 17 (Lecture 45):

Motorola 68K processor:

68K processors have 16 23-bit general purpose registers named from A0-A7 and D0-D7.

A0-A7 can hold addresses in indirect memory accesses. These can also be used as software stack pointers. Stack in 68K is not as rigid a structure as it is in x86.

Registers:

There is a 32-bit program counter (PC) that holds the address of currently executing instruction. The 8-bit condition code register (CCR) holds the X (Extend) N (Negative) Z (Zero) V (Overflow) C (Carry) flags. X is set to C for extended operations (addition, subtraction, or shifting).

Data Types:

Motorola processors allow bit addressing, that is a specific bit in a byte or a bit field, i.e. a number of bits can be directly accessed. This is a very useful feature especially in control applications. Other data types include byte, word, long word, and quad word. A special MOVE16 instruction also accepts a 16-byte block.

Memory Access:

68K allows indirect memory access using any A register. A special memory access allows post increment or pre-decrement as part of memory access. These forms are written as (An), (An)+, and –(An). Other forms allow addressing with another register as index and with constant displacement.

Using one of the A registers as the stack pointer and using the post increment and pre-decrement forms of addressing, stack is implemented. Immediate can also be given as arguments and are preceded with a hash sign (#).

Addressing is indicated with parenthesis instead of brackets. 68K has no segmentation; it however has a paged memory model.

Byte Order:

It used the big-endian format in contrast to the little endian used by the Intel processors. It has varying instruction lengths from 1-11 words. It has a decrementing stack just like the Intel one.

Motorola Instructions:

The format of instructions is **“operation source, destination”** which is different from the Intel order of operands.

👉 Also Recommended: CS401 SHORT NOTES CHAPTER 1-7 Midterm

Some instructions from various instruction groups are given below.

Data Movement

EXG D0, D2
MOVE.B (A1), (A2)

MOVEA (2222). L, A4
MOVEQ #12, D7

Arithmetic

ADD D7, (A4)
CLR (A3) (set to zero)
CMP (A2), D1
ASL, ASR, LSL, LSR, ROR, ROL, ROXL, ROXR (shift operations)

Program Control

BRA label
JMP (A3)

BSR label	(CALL)
JSR (A2)	(indirect call)
RTD #4	(RET N)

Conditional Branch:

BCC	(branch if carry clear)
BLS	(branch if Lower or Same)
BLT	(branch if Less Than)
BEQ	(branch if Equal)
BVC	(branch if Overflow clear)

SUN SPARC Processor:

The Sun SPARC is a very popular processing belonging to the RISC (reduced instruction set computer) family of processors. RISC processors originally named because of the very few rudimentary instructions they provided, are now providing almost as many instructions as CISC (complex instruction set computer). However, some properties like a fixed instruction size and single clock execution for most instructions are there.

SPARC stands for **S**calable **P**rocessor **A**rchitecture. SPARC is a 64-bit processor.

👉 Also Recommended: CS401 SHORT NOTES CHAPTER 1-7 Midterm

Byte Order:

Its byte order is user settable and even on a per program basis. So, one program may be using little endian byte order and another may be using big endian at the same time.

Data Types:

Data types include byte, Halfword, Word (32bit), and Double Word (64bits) and Quadword. It has a fixed 32bit instruction size. It has a concept of ASI (Address Space Identifier); an 8-bit number that works similar to a segment.

Registers:

There are 8 global registers and 8 alternate global registers. One of them is active at a time and accessible as g0-g7. Apart from that it has 8 in registers (i0-i7), 8 local registers (l0-l7), and 8 out registers (o0-o7). All registers are 64bit in size. The global registers can also be called r0-r7, in registers as r8- r15, local registers as r16-r23, and out registers as r24-r31

Registers windows:

SPARC introduces a concept of register window. One window is 24 registers and the active window is pointed to by a special register called Current Window Pointer (CWP). The actual number of registers in the processor is in hundreds not restricted by the architecture definition. Two instruction SAVE and RESTORE move this register window forward and backward by 16 registers.

Therefore, one SAVE instruction makes the out register the in registers and brings in new local and out registers. A RESTORE instruction makes the in registers out registers and restores the old local and in registers. This way parameters passing and returning can be totally done in registers and there is no need to save and restore registers inside subroutines.

The register o6 is conventionally used as the stack pointer. Return address is stored in o7 by the CALL instruction. The register g0 (r0) is always 0 so loading 0 in a register is made easy. SPARC is a totally register based architecture, or it is called a **load-store architecture** where memory access is only allowed in data movement instruction. Rest of the operations must be done on registers.

SPARC instructions:

SPARC instructions have two sources and a distinct destination. This allows more flexibility in writing programs.

Some examples of instructions of this processor follow.

👉 Also Recommended: CS401 SHORT NOTES CHAPTER 1-7 Midterm

Data Movement

LDSB [rn], rn	(load signed byte)
LDUW [rn], rn	(load unsigned word)
STH [rn], rn	(store half word)

Arithmetic

source1 = rn
source2 = rn or simm13

dest = rn
ADD r2, r3, r4
SUB r2, 4000, r5

SLL, SRA, SRL	(shifting)
AND, OR, XOR	(logical)

Program Control

CALL	(direct call)
JMPL	(register indirect)

RET

SAVE

RESTORE

BA label	(Branch Always)
BE label	(branch if equal)
BCC label	(branch if carry clear)
BLE label	(branch if less or equal)
BVS label	(branch if overflow set)

SUPER STAR WebTech

👉 Also Recommended: CS401 SHORT NOTES CHAPTER 1-7 Midterm