ELEC 464 : MICROCOMPUTER SYSTEM DESIGN
FALL 1995

Introduction to the Intel 8088 Architecture and Instruction Set

The Intel 8088 is the processor that was used as basis of the popular IBM PC and the many compatible machines. This lecture compares the 8088 architecture and instruction set with the 68000.
While it's not possible to become an expert on the 8088 in one lecture you should learn enough about the 8088 to be able to explain the major differences between it and the 68000 and to be able to write simple programs in 8088 assembly language.

Introduction

History

The original Intel 16-bit CPU was the 8086. It was designed to be backwards-compatible at the assembler level with Intel's 8-bit CPU, the 8080. The 8088 is a version of the 8086 with an 8-bit data bus. Later versions of the 8086 included the 80286 which was capable of addressing 16 MBytes.

Architecture

The 8088 architecture is simpler and less 'regular' than the 68000's. There is no supervisor mode, all features are always available.

Data Registers

Like the 68000, the 8088 has data registers that are be used for arithmetic and logic operations. The 8088 has four 16-bit data registers called AX, BX, CX and DX. The MS and LS bytes of each register can also be addressed separately (e.g. AH or CL).

Address Registers

The 8088 also has four address registers, SP (stack pointer), BP (frame base pointer), and SI and DI (source and destination index pointers). BX can also be used as an address register.

Since the address registers are only 16 bits wide, they are always used together with with one of four the segment registers CS (code segment), DS (data segment), SS (stack segment) or ES (extra segment) to form a 20-bit address. This is done by shifting the segment register value left by 4 bits before adding it to the address register.

Exercise: If CX contains 1122H, SP contains 1234H, and SS contains 2000H, what memory values will change when the PUSH CX instruction is executed?

The use of segment registers reduces the size of pointers to 16 bits. This reduces the code size but also restricts the addressing range of a pointer to 64k bytes. Performing address arithmetic within data structures larger than 64k is awkward. This is probably the biggest drawback of the 8088 architecture.

Processor Status Word

The 8088 has a 16-bit register similar to the 68000's SR. The PSW contains flags that indicate the CPU state such as interrupt masking and the result of the previous arithmetic or logic operation. Flags include sign (SF), zero (ZF), parity (PF), carry (CF) and interrupt enable (IF).

Instruction Set

Unlike the 68000's fixed-length 16-bit instructions, 8088 instructions are a minimum of 1 byte with up to 5 optional additional bytes and several optional opcode prefixes.

Assembly Language

The 8088 assembly language syntax is similar to the 68000's with some annoying differences:

- the operand format is ¡destination¿,¡source¿, e.g. `MOV AX,13H` loads AX with 19.
- the assembler keeps track of the type of each symbol and uses it select immediate or direct addressing, there is no special symbol (#) to indicate immediate addressing.
- hex and binary constants are indicated by using an H or B suffix after the number. If

the first digit of a hex constant is not number a leading 0 must be added.

- square brackets `[]` are used to indicate indirect addressing (and only a few of the registers can be used for this)

Exercise: The following table shows the 8088 opcodes for the major groups of instructions. What would be the corresponding 68000 instruction?

Move	
MOV	
Arithmetic/Logic	
ADD	
SUB	
AND	
CMP	
OR	
NOT	
XOR	
ROL	
SHL	
SAL	
RCL	
CLI	
STC	
NOP	
Branch	
JZ	
JNZ	
JLE	
JMP	

Call/Return

CALL and RET are the equivalents of JSR and RTS on the 68000.

The CALL instruction can push just the IP value (for calls within the same segment) or both IP and CS. This means it is necessary to know what calling convention was used when the routine was called. This is the reason why routines need to be declared as NEAR or FAR.

The INT and IRET are the equivalents of the TRAP and RTE 68000 instructions.

The PUSH and POP instructions work like the 68000 MOVE instructions with predecrement and post-increment addressing modes but only work with the SP register.

Addressing Modes

Unlike the 68000 where different addressing modes can usually be used with any data or address register, the 8088 only allows a few specific registers be used for each addressing mode.

The effective address can make use the use of SI or DI registers, the BX register and/or a constant displacement.

For example, MOV AX,[BX].4 copies the word at 4 bytes past the contents of BX to AX. The instruction MOV AL,TABLE[SI] copies the byte at an offset of SI past the memory location TABLE into AX. In both cases the DS register defines the segment value.

Each address register has a default segment register used with it (e.g. SI uses DS, DI uses ES, PUSH uses SS, JUMP uses CS, etc). A special prefix instruction can be used to override the default segment register for the following instruction.

Interrupt Handling

Like the 68000 the 8088 reserves the lower 1024 bytes for an interrupt vector table.

The 8088's interrupt acknowledge cycle is actually 2 read cycles. During the second read cycle it reads the interrupt number from the LS 8 bits of the data bus.

Like the 68000 some of these vectors are reserved for specific exceptions such as divide by zero (vector 0).

The 8088 can also generate any of the 255 interrupts by using a the INT instruction. This "software interrupt" is typically used to access operating system services in a fashion similar to the 68000's TRAP instruction. However, there is no change in processor state similar to the 68000's supervisor mode.

Exercise: MS-DOS programs use the INT 21H instruction to request operating system services. Where would the address of the entry point to these DOS services be found?

Like the 68000, an interrupt causes the PSW as well as the address of the next instruction to be stored on the stack. The stacked address includes the CS and IP (program counter) values.