

CSCE 313

Embedded Systems Programming

2003/1/31

Spring 2003 – Lecture 8

M68000 Address Modes

© 2003 Ms. Padmaja Sunkara - Lecturer

68000 ADDRESSING MODES

The method of locating operands

Ex: MOVE source destination

Instructions for 68000 contain two kinds of information.

- Type of function to be performed
- Location of operands on which to perform that function

68000 supports 13 addressing modes

68000 Mode – Data Register Direct

1. **Data Register direct** addressing mode identifies a data register(D0-D7) as the source or destination of data.

MOVE.B D0 D3

Before	D0	10204FFF
	D3	1034F88A

After	D0	10204FFF
	D3	1034F8FF

==

Lower byte (0..7bits) of destination register are affected.



68000 Mode – Address Register Direct

2. **Address Register direct** addressing mode moves data to an address register
Only word or longword operands may be specified

MOVEA.L A3 A0

Before	A0	00200000
	A3	0004F88A

After	A0	0004F88A
	A3	0004F88A



68000 Mode – Address Register Direct

Word is default

MOVEA A3 A0

Before A0 00200000
A3 0004F88A

After A0 FFFFFFF88A
A3 0004F88A

MSB of source is 1, therefore sign extended 32 bit has bits 16-31 equal to 1



68000 Mode – Absolute Addressing

3. Absolute Addressing

The source or destination is a memory location whose address is specified in one or two extension words of the instruction. There are two variations of absolute addressing: absolute short and absolute long.

(1) Absolute short

A single extension word is appended to the instruction to provide bits 0-15 of address. The upper bits are formed when the instruction is executed by sign extending bit 15 of the extension word to bits 16-31.



68000 Mode – Absolute Addressing

This mode may be explicitly set by appending “.s” or “<”

MOVE.B #5,\$F00.S

MOVE.B #5,<\$F00

Ex: MOVE.L #\$1E, \$800

	address	contents
Before:	0800	12
	0801	34
	0802	56
	0803	78
After :	0800	00
	0801	00
	0802	00
	0803	1E



© 2002 Dr. James P. Davis Page 7

68000 Mode – Absolute Addressing

(2) Absolute Long

Include two extension words specifying a 32 bit address of a source or destination operand. this mode may be explicitly set by appending “.l” or “>”

MOVE.B #5,\$F00.L

MOVE.B #5,>\$F00

Ex: MOVE.B #\$1E,\$8F000

	address	contents
Before:	08F000	FF
After :	08F000	1E---OPERAND SIZE IS BYTE



© 2002 Dr. James P. Davis Page 8

68000 Mode – Register Indirect

4. In **Register Indirect** addressing , an address register contains the address of the source or destination operand.

MOVE.L D0,(A0)

address contents	Registers
Before:001000 55	A0 00001000
001001 02	D0 1043834F
001002 3F	
001003 00	
After: 001000 10	A0 00001000
001001 43	D0 1043834F
001002 83	
001003 4F	



68000 Mode – Post-increment Register Indirect

5. Post-Increment Register Indirect

Read or write of data takes place before address register is incremented .

byte-1

word-2

long-4

Ex: MOVE.W (A5)+,D0

address contents	Registers
Before: 001000 45	A5 00001000
001001 67	D0 0000FFFF
001002 89	
001003 AB	
After: 001000 45	A5 00001002
001001 67	D0 00004567
001002 89	
001003 AB	



68000 Mode – Pre-decrement Register Indirect

6. Predecrement Register Indirect

The value of stack pointer is decremented by two before move takes place

MOVE.W D0,-(A7)

	address contents	Registers
Before:	001000 10	A7 00001002
	001001 12	D0 00000143
	001002 83	
	001003 47	
After:	001000 01	A7 00001000
	001001 43	D0 00000143
	001002 83	
	001003 47	



© 2002 Dr. James P. Davis Page 11

68000 Mode – Register Index + Offset

7. Register with Index and offset

EA=sign extended offset+contents of index register + contents of address register.

MOVE.W \$10(A0,D0.L),A1

	address contents	Registers
Before:	00101C EF	A0 0000100A
	00101D 10	A1 00000000
		D0 00000002
After:	00101C EF	A0 0000100A
	00101D 10	A1 FFFFEF10
		D0 00000000

EA=\$10+\$100A+\$2=\$101C



© 2002 Dr. James P. Davis Page 12

68000 Mode – Register Indirect + Offset

8. Register Indirect with offset

The sign extended offset is added to the address register to form the effective address of source or destination

```
MOVE.W 6(A0),D0
```

	address contents	Registers
Before:	001000 07 001027 BF	A0 00001020 D0 00000000
After:	001026 07 001027 BF	A0 00001020 D0 000007BF



68000 Mode – Program Counter Relative + Offset

9. PC Relative with offset

This addressing mode uses a 16 bit offset which is added to the PC to form the effective address.

```
MOVE.W $1020(PC),D5
```

	address contents	Registers
Before:	001020 AB 001021 CD	PC 00001000 D5 12345678
After:	001020 AB 001021 CD	PC 00001004 D5 1234ABCD



68000 Mode – PC Relative + Index + Offset

10. PC Relative with Index and Offset

This addressing mode previous except the offset is limited to 8 bits and an index register also used in computing the EA.

```
MOVE.W $1020(PC,D0.W),D5
```

	address contents	Registers
Before:	001026 FE	PC 00001000
	001027 DC	D0 ABCD0006
		D5 12345678
After:	001026 FE	PC 00001004
	001027 DC	D0 ABCD0006
		D5 1234FEDC

$20+00+06=026$

W-----INDICATES ONLY LOWER ORDER 16 BITS ARE BEING USED.



68000 Mode – Immediate Addressing

11. **Immediate Addressing** uses one or two extension words to hold source operand

```
MOVE.L #$1FFFF,D0
```

	Registers	contents
Before:	D0	12345678
After:	D0	0001FFFF

Base \$=hexadecimal

@=octal

%=binary

= means immediate (i.e. constant) data follows.

A variation of immediate addressing is quick immediate.

```
MOVEQ #$1F,D0
```

```
ADDQ
```

```
SUBQ
```



68000 Mode – Implied Register Addressing

12. Implied Register

A few instructions implicitly use a register as a source or destination operand or in computing the effective address for branching.

MOVE.B #1,CCR

Sets bit 0, the carry bit in the condition code register and clears the other bits. The destination addressing mode is implied register since no field is present in instruction word identifying the CCR. It is implicitly encoded in the instruction word.



68000 Mode –Relative Addressing

13. Relative Addressing

The 68000 uses several variations of branch instructions that use relative addressing.

BRA - branch always

e.g., BRA LOOP

Bcc - branch conditional

e.g., BGT, BLE, etc.

BSR - branch to subroutine

e.g., BSR FOOBAR

DBcc - test condition, decrement and branch

e.g., DBLE END



68000 Mode –Relative Addressing (example)

1	00002000	ORG \$2000	
2	00002001 6000001E	BRA AHEAD	┌
3	00002020	ORG \$2020	└
4	00002020 4E71 AHEAD	NOP	← Jump here!!!

How it works (loosely):

The Linker makes the appropriate calculations of the relative offsets, which are then incorporated into the machine instruction. This value is encountered by the Instruction Decoder unit of the CPU, which carries out the relative reference, passing it to the Fetch Unit to set the PC to the new location. The next fetch would take place from that location.

