Chapter 6

Address Register

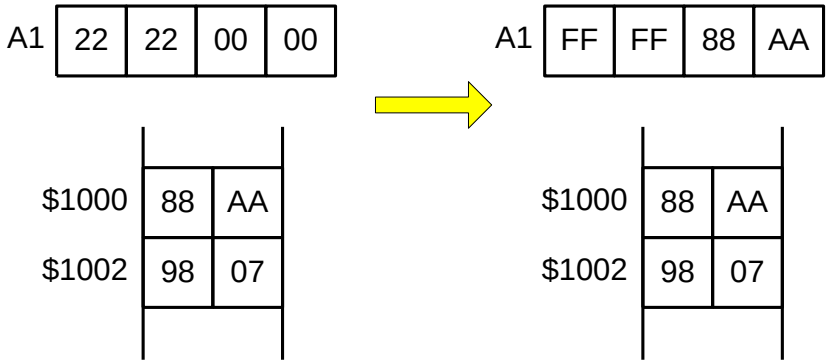
Expected Outcomes

- Differentiate between direct and indirect register modes
- Compare and contrast the various modes of ARI operation; ARI with pre-decrement, post-increment, ARI with displacement and ARI with index and displacement
- Analyze the result of operation for different ARI modes

Address Register Direct

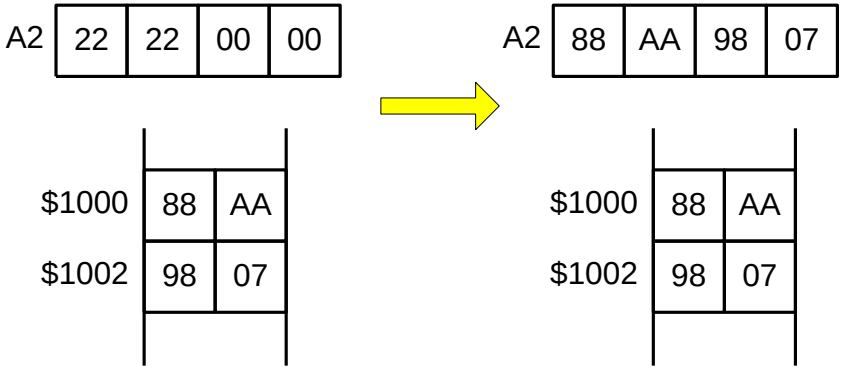
- To access the content of address register
- Normally, required special instruction and all the content of the address register are changed
- The CCR is unchanged

- **Example**: `MOVEA.W $1000,A1`



Address Register Direct

- **Example:** `MOVEA.L $1000,A1`



- **Example:** `MOVEA.W #$7000,A1`

Address Register Direct

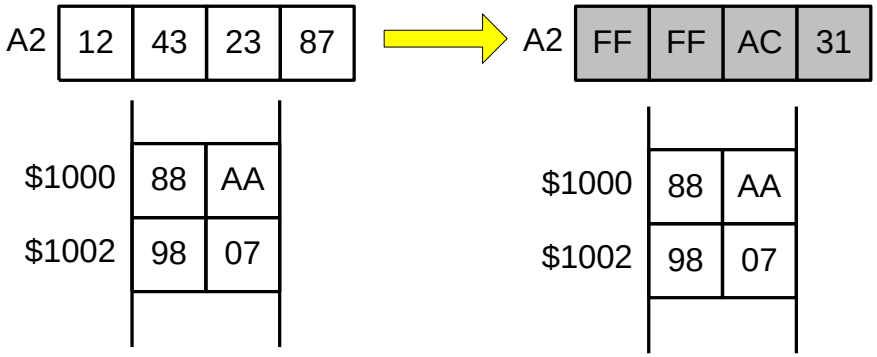
- **Example:** `MOVEA.L #$7000,A1`

- **Example:** `MOVEA.W #$9000,A1`

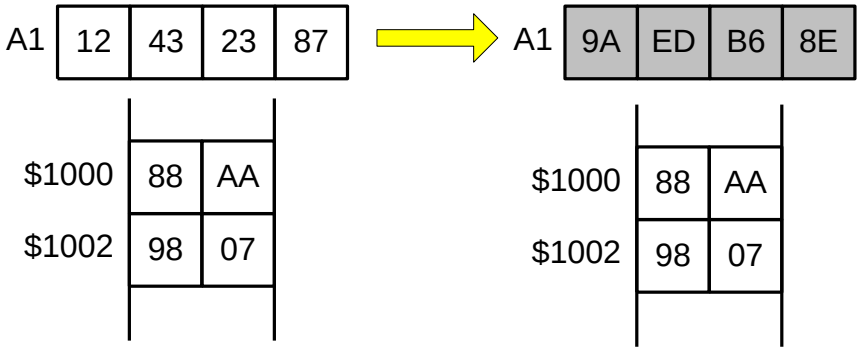
- **Example:** `MOVEA.L #$9000,A1`

Address Register Direct

- **Example:** `ADDA.W $1000,A2`



- **Example:** `ADDA.L $1000,A1`

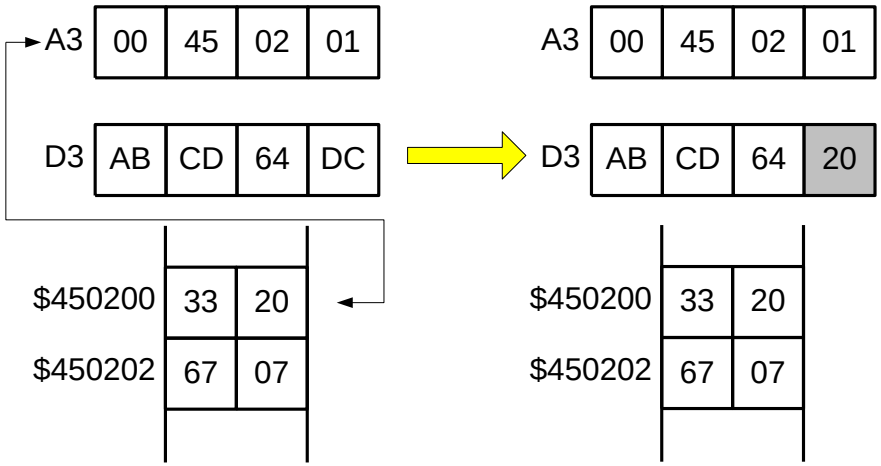


Address Register Direct

- **Example:** `SUBA.W  #$9ABC,A3`

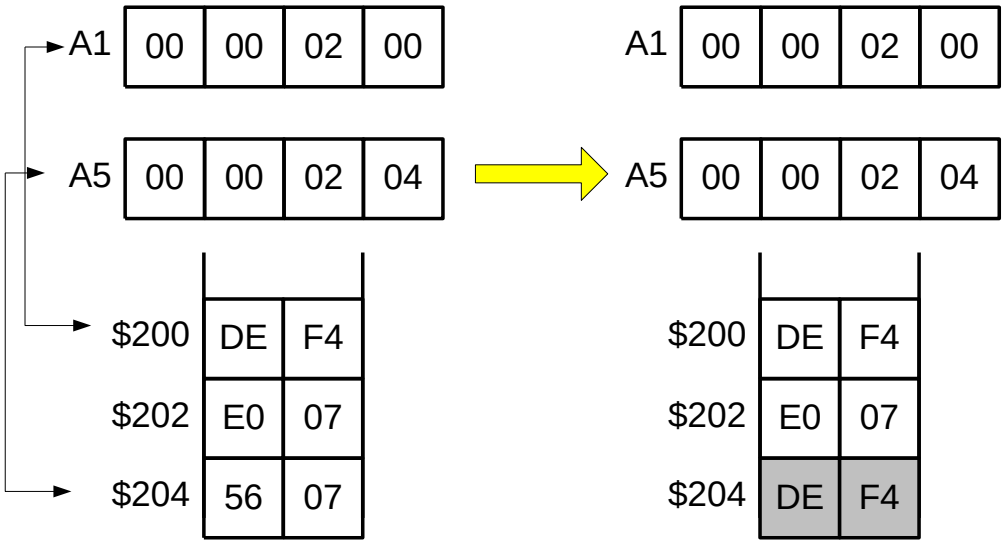
Address Register Indirect

- The address register indirect (ARI) mode is used to access memory through address register
- The content of address register is the effective address of the data
- **Example:** `MOVE.B (A3),D3`



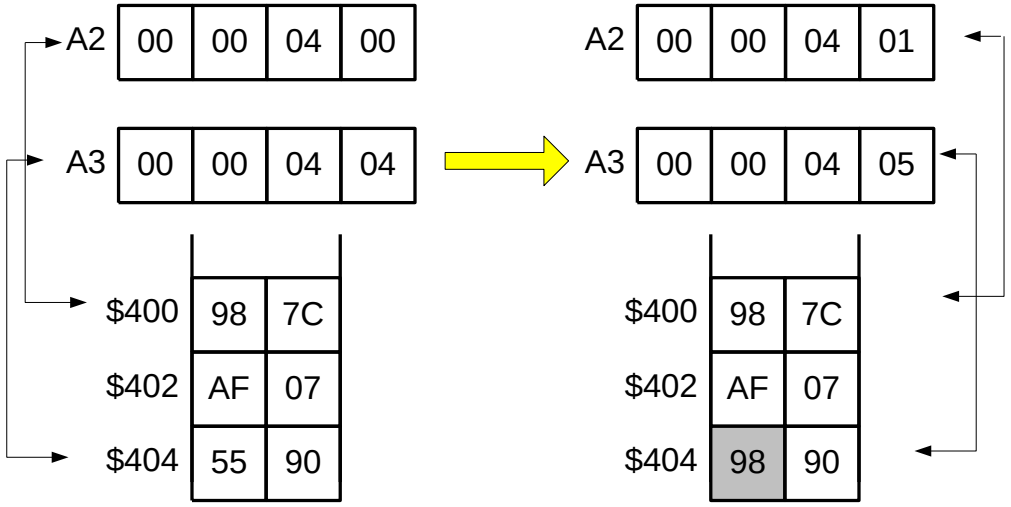
Address Register Indirect

- **Example:** `MOVE.W (A1),(A5)`



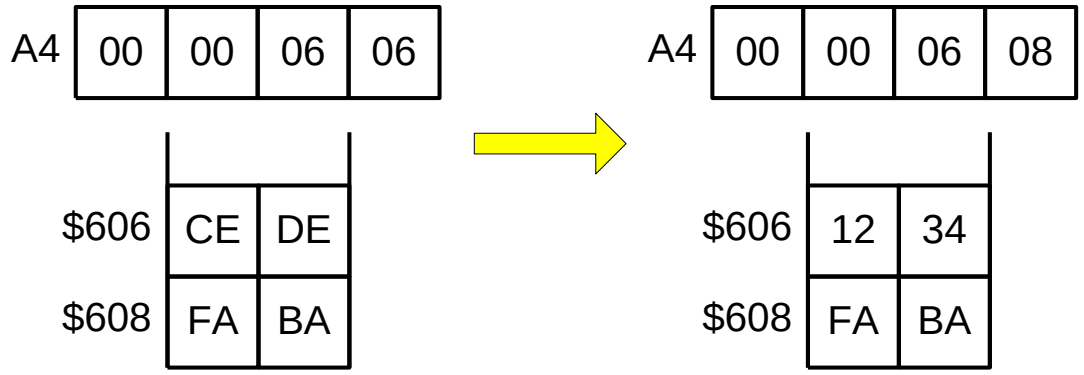
ARI with Post-Increment

- The value incremented depends on the size of operation
- Like the ARI technique but the value of address register is incremented after it is used
  - 1 for byte, 2 for word and 4 for longword
- **Example:** `MOVE.B (A2)+,(A3)+`



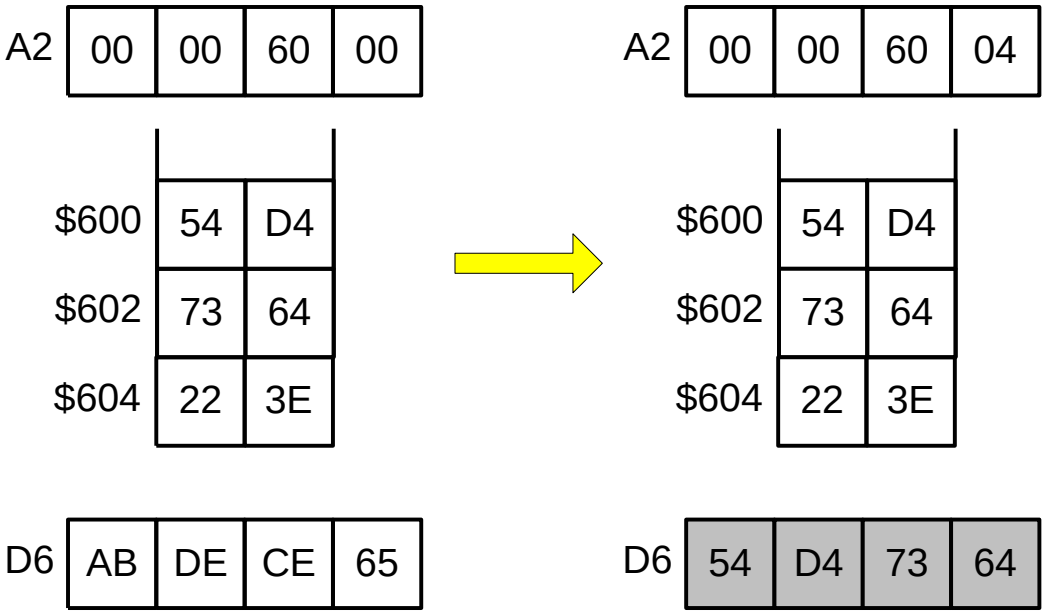
ARI with Post-Increment

- **Example:** `MOVE.W #1234,(A4)+`



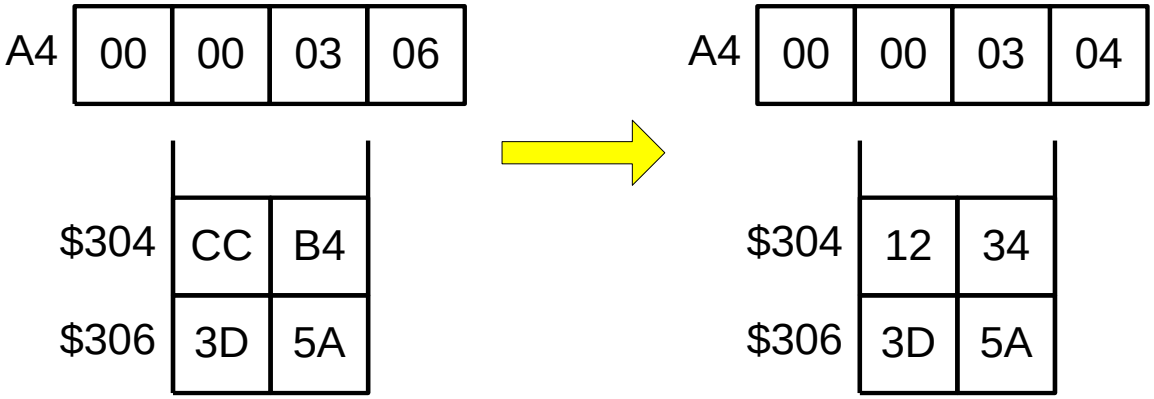
ARI with Post-Increment

- **Example :** `MOVE.L (A2)+,D6`



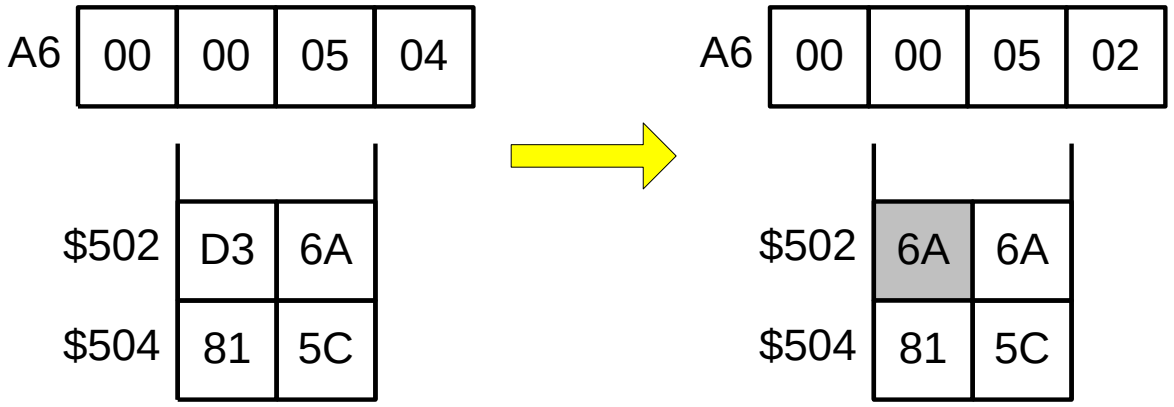
ARI With Pre-Decrement

- Like the ARI technique but the value of address register is decremented before it is used
- The value decremented depends on the size operation
  - 1 for byte, 2 for word and 4 for longword
- **Example:** `MOVE.W #$1234,-(A4)`



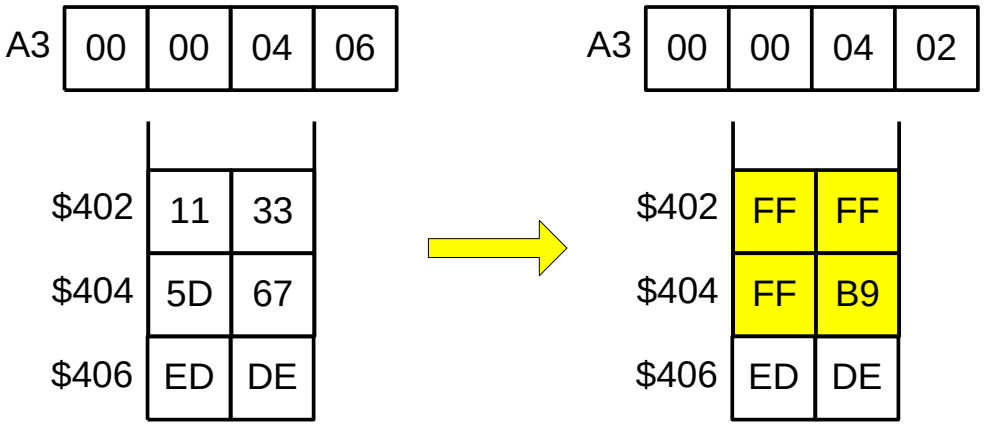
ARI With Pre-Decrement

- **Example:** `MOVE.B -(A6),-(A6)`



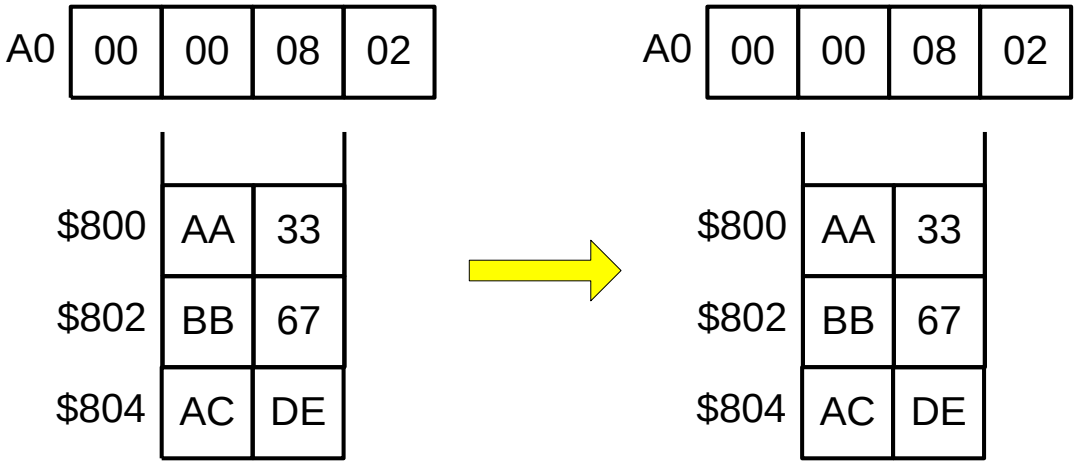
ARI With Pre-Decrement

- **Example:** `MOVE.L #-$47,-(A3)`



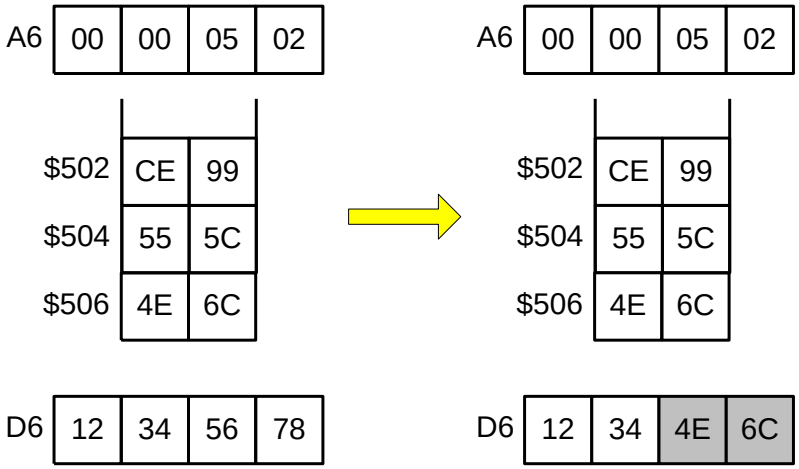
ARI With Pre-Decrement

- **Example:** `MOVE.W (A0)+,-(A0)`



ARI With Offset

- The effective address is determined by adding the content of selected address register with the offset, that is
  Effective address (EA) = $An+d_{16}$
- The offset is 16-bit signed integer, thus ranging from -32768 to +32767
- The ARI with Offset format : $d_{16}(An)$
- **Example:** `MOVE.W 4(A6),D6`



ARI With Index & Offset

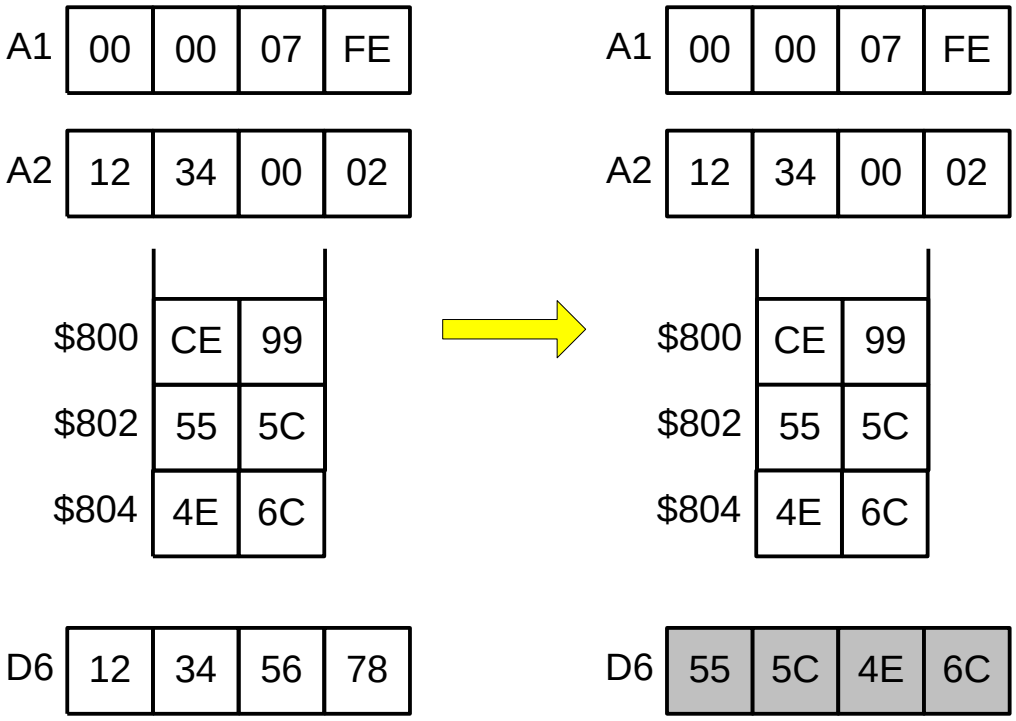
- The effective address is determined by adding the content of selected address register with the content of index register and offset, that is

  Effective address (EA)= $An + IX + d_8$

- The offset size is 8-bit signed integer, thus ranging from -128 to +127
- The index register can be either address register (An) or data register (Dn) and it can be in a word or a longword form
- The ARI with Offset format : $d_8(An,IX.s)$

ARI With Index & Offset

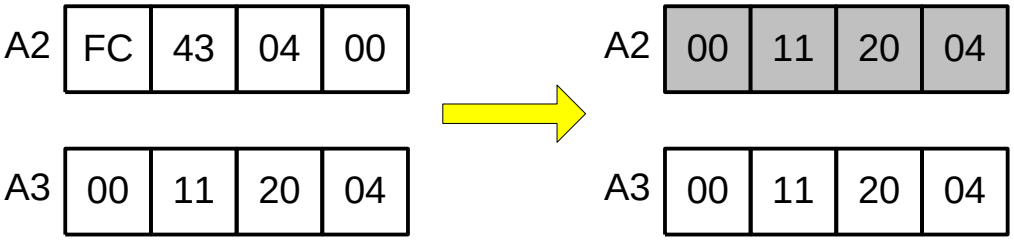
- **Example**: `MOVE.L 2(A1,A2.W),D6`



LEA Instruction

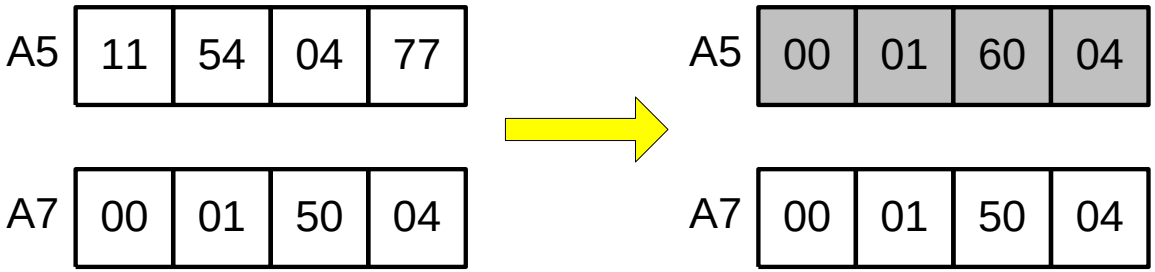
- Calculate the effective address and fill in address register
- 32 bit size operation
- **Example:** `LEA $30000,A2`

- **Example:** `LEA (A3),A2`



LEA Instruction

- **Example:** `LEA $1000(A7),A5`



Self-Test

Exercise

The content of various registers and memories are

```
A3=$00004002,A6=$00004010,PC=$00001000
($004000)=$FF112233 44556677 889900AA BBCCDDEE
```

Determine the value of each content if the program is executed

```
ORG         $7000
MOVEQ       #-2,D0
MOVE.L      -(A6),-(A6)
MOVE.W      (A3)+,(A3)+
MOVE.W      6(A3),$4000
MOVE.L      0(A3,D0),(A6)+
MOVE.W      (A3)+,2(A6)
```

Self-Test

Exercise

The content of various registers and memories are

```
A0=$00004002,A1=$00004010,D0=$00000004
```

Determine the content of address $4000 - $4010 if the program is executed

```
ORG         $2000
MOVEQ       #-2,D0
MOVE.B      -(A1), -(A1)
MOVE.B      (A0)+,(A0)+
MOVE.B      2(A0),$4000
MOVE.B      2(A1,D0),(A0)+
MOVE.B      D0,6(A0)
ORG         $4000
DC.B        'WELCOME TO FKEE,UMP'
```

Self-Test

Exercise

The content of various registers and memories are,

```
A0=$00003004,D0=$12340004,D1=$24689753,A1=$00003006
($3000)=1122 3344 5566 7788 9900 AABB CCDD EEFF FEED
```

Evaluate the following instruction and obtain the new value of D0 and D1

Instruction	D0	D1
ADD.B (A0),D0		
SUB.W 4(A1),D1		
MOVE.L 2(A0,D0),D1		
MULS -(A1),D1		

Self-Test

Exercise

Determine the clock cycle, bytes requirement and machine code for each instruction

Instruction	Clock cycle	# of bytes	Machine Code
`CLR.L (A0)`			
`CLR.W 5(A2,A1)`			
`MOVE.L $40000,10(A6)`			
`MOVE.B (A0)+,$10000`			
`MOVE.L -(A3),2(A0,D1)`			
`ADD.B (A4)+,D3`			

Self-Test

Exercise

If the content of registers are as follow,

```
A0=$00012000,A1=$00008000,A2=$00020000
D0=$00004000
```

obtain the content of register for each instruction

Instruction	A0	# of bytes	Machine Code
LEA $1000,A0			
LEA (A1),A0			
LEA -$6000(A2),A0			
LEA -$10(A0,D2.W),A0			