Two-Dimensional Arrays

Ordering of Rows and Columns

- Row-Major Order

  When row-major order (most common) is used, the first row appears at the beginning of the memory block. The last element in the first row is followed in memory by the first element of the second row.

- Column-Major Order

  When column-major order is used, the elements in the first column appear at the beginning of the memory block. The last element in the first column is followed in memory by the first element of the second column.

Row-Major and Column-Major Ordering

Logical arrangement:

10	20	30	40	50
60	70	80	90	A0
B0	C0	D0	E0	F0

Row-major order

10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0

Column-major order

10	60	B0	20	70	C0	30	80	D0	40	90	E0	50	A0	F0

Operand Types

- Base-Index Operands
- Base-Index-Displacement Operands

Base-Index Operands

- A base-index operand adds the values of two registers(called base and index), producing an offset address:

  [base + index]

```
.data
array WORD 1000h,2000h,3000h
.code
mov   ebx,OFFSET array
mov   esi,2
mov   ax,[ebx+esi]              ; AX = 2000h

mov   edi,OFFSET array
mov   ecx,4
mov   ax,[edi+ecx]              ; AX = 3000h

mov   ebp,OFFSET array
mov   esi,0
mov   ax,[ebp+esi]              ; AX = 1000h
```

Base-Index Operands

- When accessing a two-dimensional array in row-major order, the row offset is held in the base register and the column offset is in the index register.

```
tableB  BYTE   10h,  20h,  30h,  40h,  50h
Rowsize = ($ - tableB)
        BYTE   60h,  70h,  80h,  90h, 0A0h
        BYTE  0B0h, 0C0h, 0D0h, 0E0h, 0F0h
```

Base-Index Operands

- Suppose we want to locate a particular entry in the table using row and column coordinates.
- Assuming that the coordinates are zero based, the entry at row 1, column 2 contains 80h.
- We set EBX to the table's offset, add (Rowsize * row index) to calculate the row offset, and set ESI to the column index.

Base-Index Operands

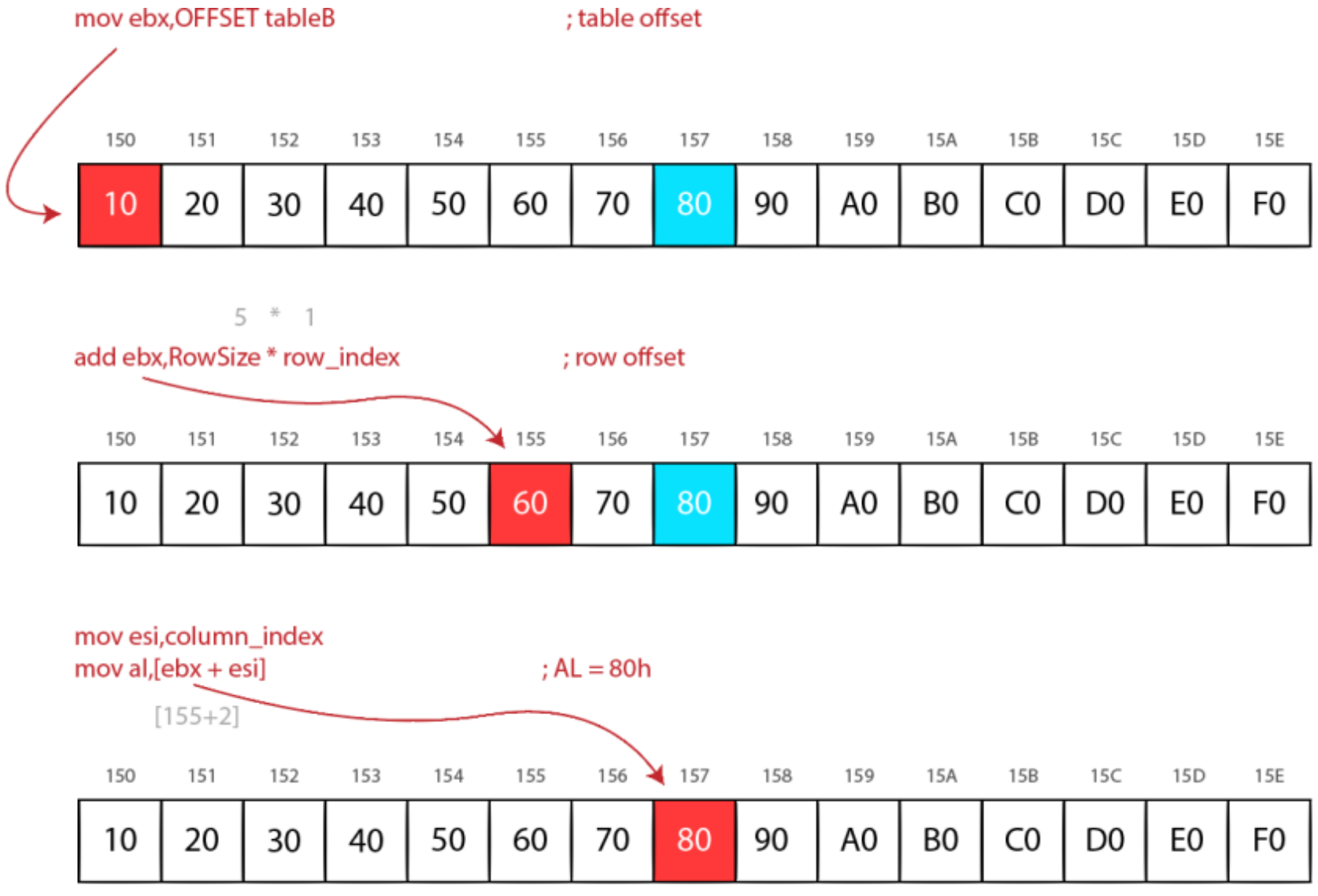
- Suppose the array is located at offset 0150h.

row_index = 1
column_index = 2
Rowsize = 5

	0	1	2	3	4
0	10	20	30	40	50
1	60	70	80	90	A0
2	B0	C0	D0	E0	F0

150	151	152	153	154	155	156	157	158	159	15A	15B	15C	15D	15E
10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0

Base-Index Operands



Scale Factors

If you're writing code for an array of WORD, multiply the index operand by a scale factor of 2.

```
tableW  WORD   10h,  20h,  30h,  40h,  50h
RowsizeW = ($ - tableW)
        WORD   60h,  70h,  80h,  90h, 0A0h
        WORD  0B0h, 0C0h, 0D0h, 0E0h, 0F0h
.code
row_index = 1
column_index = 2
mov   ebx,OFFSET tableW                  ; table offset
add   ebx,RowSizeW * row_index           ; row offset
mov   esi,column_index
mov   ax,[ebx + esi*TYPE tableW]         ; AX = 0080h
```

Scale Factors

Similarly, you must use a scale factor of 4 if the array contains doublewords:

```
tableD DWORD 10h, 20h, ...etc.
.code
mov   eax,[ebx + esi*TYPE tableD]
```

Base-Index-Displacement Operands

- A base-index-displacement operand combines a displacement, a base register, an index register, and an optional scale factor to produce an effective address. Here are the formats:
  [base + index + displacement] displacement[base + index]
- Displacement can be the name of a variable.
- Base-index-displacement operands are well suitedto processing two-dimensional arrays.
- The displacement can be an array name, the base operand can hold the row offset, and the index operand can hold the column offset.

Base-Index-Displacement Operands

```
tableD DWORD   10h,  20h,  30h,  40h,  50h
Rowsize = ($ - tableD)
       DWORD   60h,  70h,  80h,  90h,  0A0h
       DWORD  0B0h, 0C0h, 0D0h, 0E0h,  0F0h
```

- Assuming that the coordinates are zero based, the entry at row 1, column 2 contains 80h. To access this entry, we set EBX to the row index and ESI to the column index:

```
mov   ebx,Rowsize                    ; row index
mov   esi,2                          ; column index
mov   eax,tableD[ebx + esi*TYPE tableD]
```

Base-Index-Displacement Operands

- How to get this value?

```
tableD DWORD   10h,  20h,  30h,  40h,  50h
Rowsize = ($ - tableD)
       DWORD   60h,  70h,  80h,  90h,  0A0h
       DWORD  0B0h, 0C0h, 0D0h, 0E0h,  0F0h
```

Base-Index-Displacement Operands

- How to get this value?

```
tableD DWORD   10h,  20h,  30h,  40h,  50h
Rowsize = ($ - tableD)
       DWORD   60h,  70h,  80h,  90h,  0A0h
       DWORD  0B0h, 0C0h, 0D0h, 0E0h,  0F0h
```

```
mov ebx,Rowsize * 2                    ; row index
mov esi,3                              ; column index
mov eax,tableD[ebx + esi*TYPE tableD]
```