마이크로프로세서

- Matrix 연산 -

Daejin Park

School of Electronics Engineering, KNU, KOREA

2019.04.12

ACES Lab.

Matrix 연산 – 메모리 할당

$$\begin{bmatrix} A0 & A1 \\ A2 & A3 \end{bmatrix}\begin{bmatrix} B0 & B1 \\ B2 & B3 \end{bmatrix} = \begin{bmatrix} C0 & C1 \\ C2 & C3 \end{bmatrix}$$

레지스터 할당

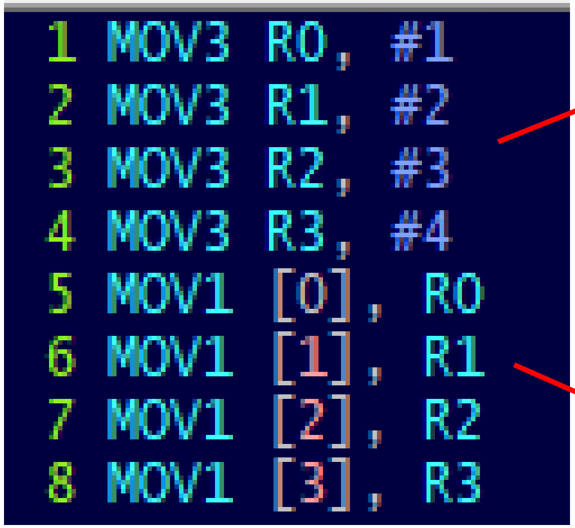
$$\begin{bmatrix} R0 & R1 \\ R2 & R3 \end{bmatrix}\begin{bmatrix} R4 & R5 \\ R6 & R7 \end{bmatrix} = \begin{bmatrix} C0 & C1 \\ C2 & C3 \end{bmatrix}$$

모든 변수는 메모리에 할당

$$\begin{bmatrix} M[0] & M[1] \\ M[2] & M[3] \end{bmatrix}\begin{bmatrix} M[4] & M[5] \\ M[6] & M[7] \end{bmatrix} = \begin{bmatrix} M[8] & M[9] \\ M[10] & M[11] \end{bmatrix}$$

값 초기화 (메모리 값 초기화)

다음의 예를 계산해보자 $\begin{bmatrix}1 & 2\\ 3 & 4\end{bmatrix}\begin{bmatrix}5 & 6\\ 7 & 8\end{bmatrix}=\begin{bmatrix}C0 & C1\\ C2 & C3\end{bmatrix}$



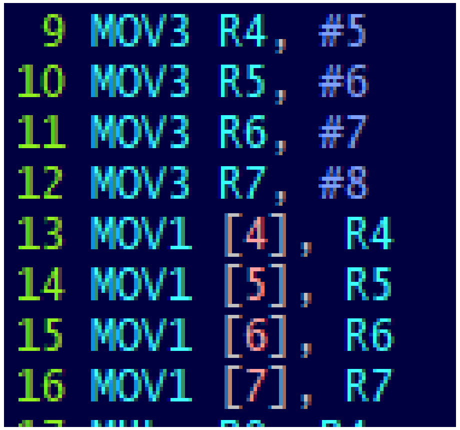
```
MOV3 R0, #1
MOV3 R1, #2
MOV3 R2, #3
MOV3 R3, #4
MOV1 [0], R0
MOV1 [1], R1
MOV1 [2], R2
MOV1 [3], R3
```

R0	R1	R2	R3	R4	R5	R6	R7
1	2	3	4				

M[0]	M[1]	M[2]	M[3]	M[4]	M[5]	M[6]	M[7]
1	2	3	4				

값 초기화 (메모리 값 초기화)

다음의 예를 계산해보자 $$\begin{bmatrix}1 & 2\\3 & 4\end{bmatrix}\begin{bmatrix}5 & 6\\7 & 8\end{bmatrix}=\begin{bmatrix}C0 & C1\\C2 & C3\end{bmatrix}$$



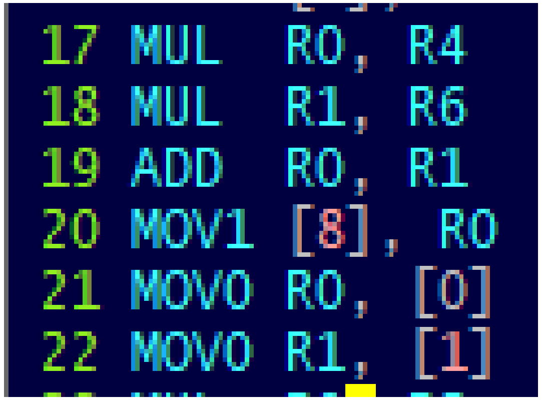
```
MOV3 R4, #5
MOV3 R5, #6
MOV3 R6, #7
MOV3 R7, #8
MOV1 [4], R4
MOV1 [5], R5
MOV1 [6], R6
MOV1 [7], R7
```

R0	R1	R2	R3	R4	R5	R6	R7
1	2	3	4	5	6	7	8

M[0]	M[1]	M[2]	M[3]	M[4]	M[5]	M[6]	M[7]
1	2	3	4	5	6	7	8

C0계산

$$\begin{bmatrix}1 & 2\\ 3 & 4\end{bmatrix}\begin{bmatrix}5 & 6\\ 7 & 8\end{bmatrix}=\begin{bmatrix}C0 & C1\\ C2 & C3\end{bmatrix}$$



```
MUL   R0, R4
MUL   R1, R6
ADD   R0, R1
MOV1  [8], R0
MOV0  R0, [0]
MOV0  R1, [1]
```

R0	R1	R2	R3	R4	R5	R6	R7
1	2	3	4	5	6	7	8

R0	R1	R2	R3	R4	R5	R6	R7
5	14	3	4	5	6	7	8

R0	R1	R2	R3	R4	R5	R6	R7
19	14	3	4	5	6	7	8

M[0]	M[1]	M[2]	M[3]	M[4]	M[5]	M[6]	M[7]
1	2	3	4	5	6	7	8

C0	C1	C2	C3
M[8]	M[9]	M[10]	M[11]
19			

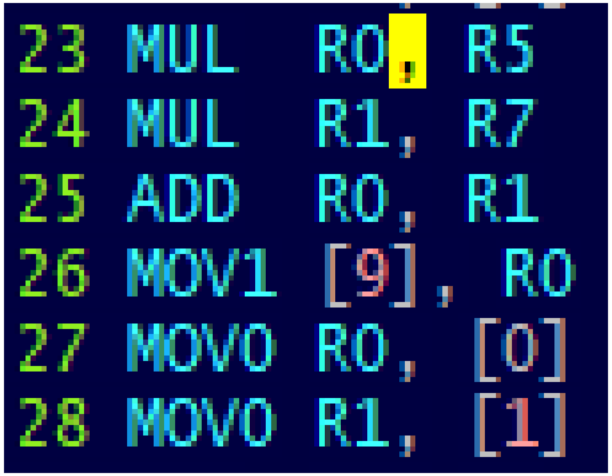
복원

R0	R1	R2	R3	R4	R5	R6	R7
1	2	3	4	5	6	7	8

ACES Lab.

C1계산

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}\begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix} = \begin{bmatrix} C0 & C1 \\ C2 & C3 \end{bmatrix}$$



```
MUL   R0, R5
MUL   R1, R7
ADD   R0, R1
MOV1  [9], R0
MOV0  R0, [0]
MOV0  R1, [1]
```

R0	R1	R2	R3	R4	R5	R6	R7
1	2	3	4	5	6	7	8

R0	R1	R2	R3	R4	R5	R6	R7
6	16	3	4	5	6	7	8

R0	R1	R2	R3	R4	R5	R6	R7
22	16	3	4	5	6	7	8

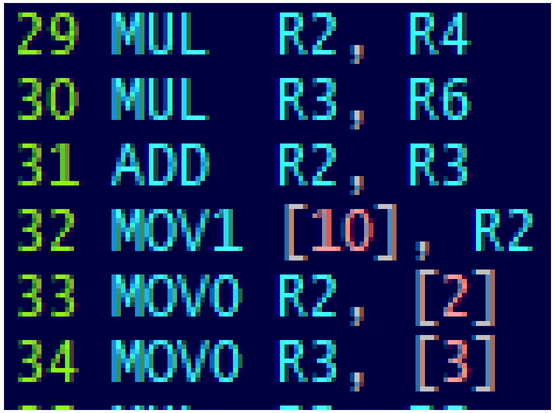
								C0	C1	C2	C3
M[0]	M[1]	M[2]	M[3]	M[4]	M[5]	M[6]	M[7]	M[8]	M[9]	M[10]	M[11]
1	2	3	4	5	6	7	8	19	22		

복원

R0	R1	R2	R3	R4	R5	R6	R7
1	2	3	4	5	6	7	8

C2계산

$$\begin{bmatrix}1 & 2\\ 3 & 4\end{bmatrix}\begin{bmatrix}5 & 6\\ 7 & 8\end{bmatrix} = \begin{bmatrix}C0 & C1\\ C2 & C3\end{bmatrix}$$



```
MUL  R2, R4
MUL  R3, R6
ADD  R2, R3
MOV1 [10], R2
MOV0 R2, [2]
MOV0 R3, [3]
```

R0	R1	R2	R3	R4	R5	R6	R7
1	2	3	4	5	6	7	8

R0	R1	R2	R3	R4	R5	R6	R7
1	2	15	28	5	6	7	8

R0	R1	R2	R3	R4	R5	R6	R7
1	2	43	4	5	6	7	8

M[0]	M[1]	M[2]	M[3]	M[4]	M[5]	M[6]	M[7]
1	2	3	4	5	6	7	8

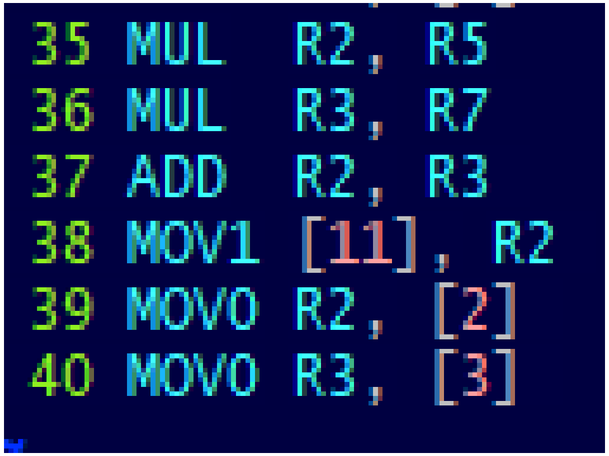
C0	C1	C2	C3
M[8]	M[9]	M[10]	M[11]
19	22	43	

복원

R0	R1	R2	R3	R4	R5	R6	R7
1	2	3	4	5	6	7	8

C3계산

$$\begin{bmatrix}1 & 2\\ 3 & 4\end{bmatrix}\begin{bmatrix}5 & 6\\ 7 & 8\end{bmatrix} = \begin{bmatrix}C0 & C1\\ C2 & C3\end{bmatrix}$$



```
MUL  R2, R5
MUL  R3, R7
ADD  R2, R3
MOV1 [11], R2
MOV0 R2, [2]
MOV0 R3, [3]
```

R0	R1	R2	R3	R4	R5	R6	R7
1	2	3	4	5	6	7	8

R0	R1	R2	R3	R4	R5	R6	R7
1	2	18	32	5	6	7	8

R0	R1	R2	R3	R4	R5	R6	R7
1	2	50	4	5	6	7	8

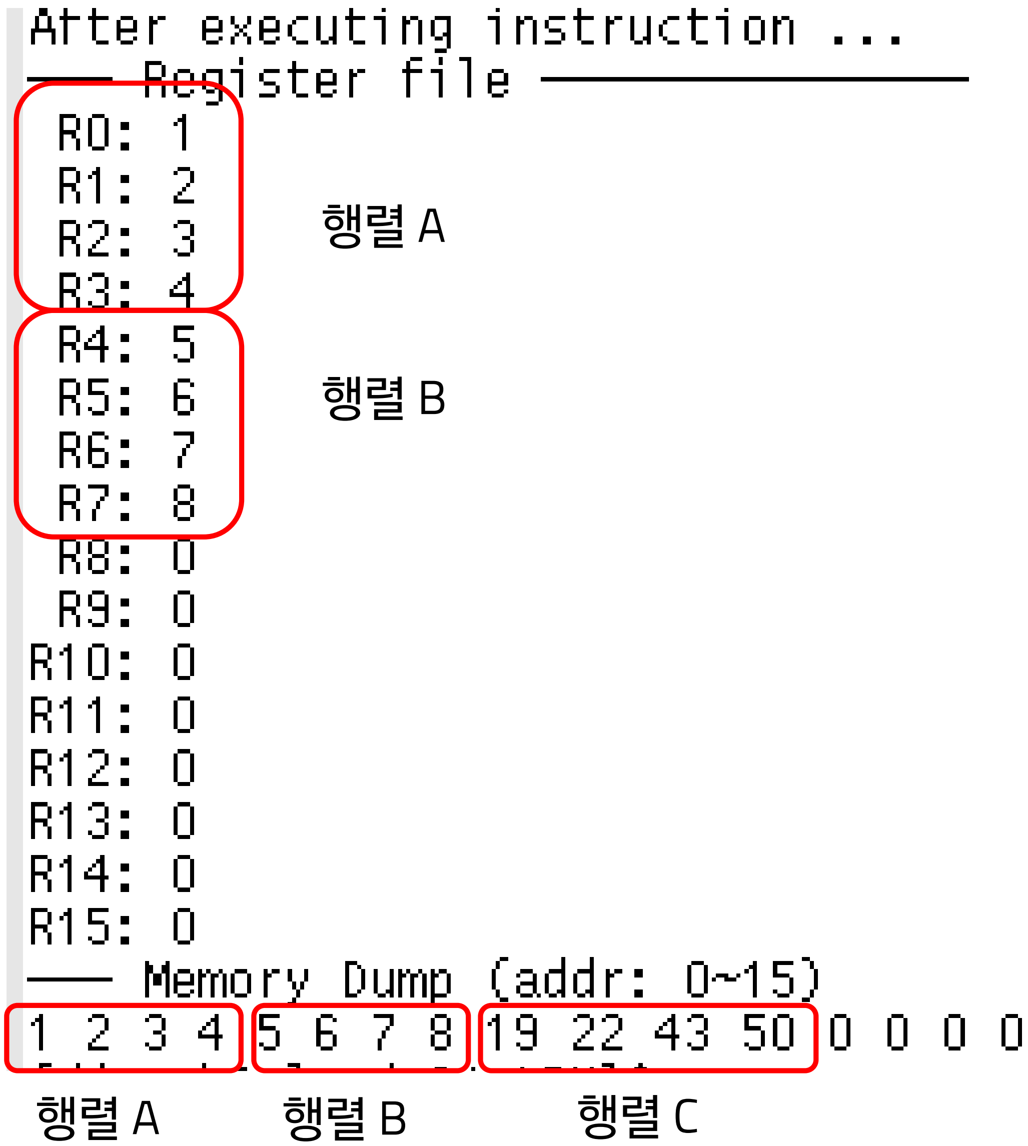
								C0	C1	C2	C3
M[0]	M[1]	M[2]	M[3]	M[4]	M[5]	M[6]	M[7]	M[8]	M[9]	M[10]	M[11]
1	2	3	4	5	6	7	8	19	22	43	50

복원

R0	R1	R2	R3	R4	R5	R6	R7
1	2	3	4	5	6	7	8

실행 결과

40줄 명령어 실행 결과



```
After executing instruction ...
—— Register file ————————
  R0:  1
  R1:  2
  R2:  3
  R3:  4
  R4:  5
  R5:  6
  R6:  7
  R7:  8
  R8:  0
  R9:  0
 R10:  0
 R11:  0
 R12:  0
 R13:  0
 R14:  0
 R15:  0
—— Memory Dump (addr: 0~15)
1 2 3 4 5 6 7 8 19 22 43 50 0 0 0 0
```

Q & A

Thank you for your attention

Architecture and Compiler
for Embedded Systems Lab.

School of Electronics Engineering, KNU

ACES Lab (boltanut@knu.ac.kr)