

정기현 교수님 컴퓨터구조 프로젝트

전자공학과 202021025 안준영

제출일: 2022/12/01

-설계 조건

- 1) Address = A0(MSB)~A19.
- 2 CS(positive trigger) 3개 사용.
- 3) $\text{mod}(202021025, 24) = 17$ 이므로, DV17부터 시작.
- 4) Date 연결 생략.
- 5) Decoder는 1x2, 2x4, 3x8만 사용 가능.

1. Hexadecimal Address map

각 메모리의 Size에 따라서 주소가 할당.

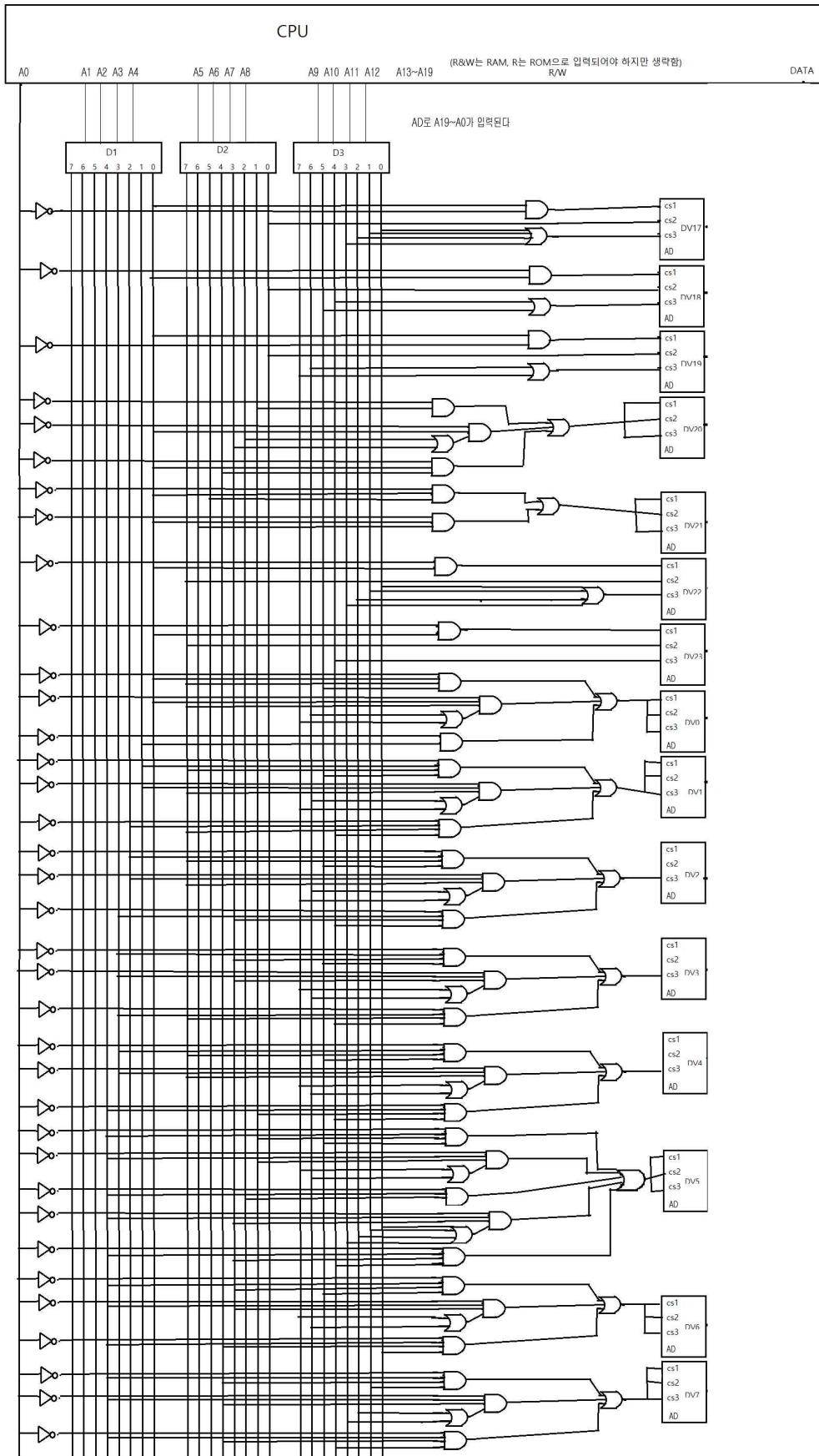
메모리	첫 주소	끝 주소
DV17	00000	00FFF
DV18	01000	017FF
DV19	01800	01FFF
DV20	02000	09FFF
DV21	0A000	0DFFF
DV22	0E000	0EFFF
DV23	0F000	0F3FF
DV0	0F400	1F3FF
DV1	1F400	2F3FF
DV2	2F400	373FF
DV3	37400	3F3FF
DV4	3F400	433FF
DV5	43400	473FF
DV6	47400	483FF
DV7	48400	493FF
DV8	49400	49BFF
DV9	49C00	4A3FF
DV10	4A400	6A3FF
DV11	6A400	8A3FF
DV12	8A400	8E3FF
DV13	8E400	923FF
DV14	92400	943FF
DV15	94400	963FF
DV16	96400	973FF

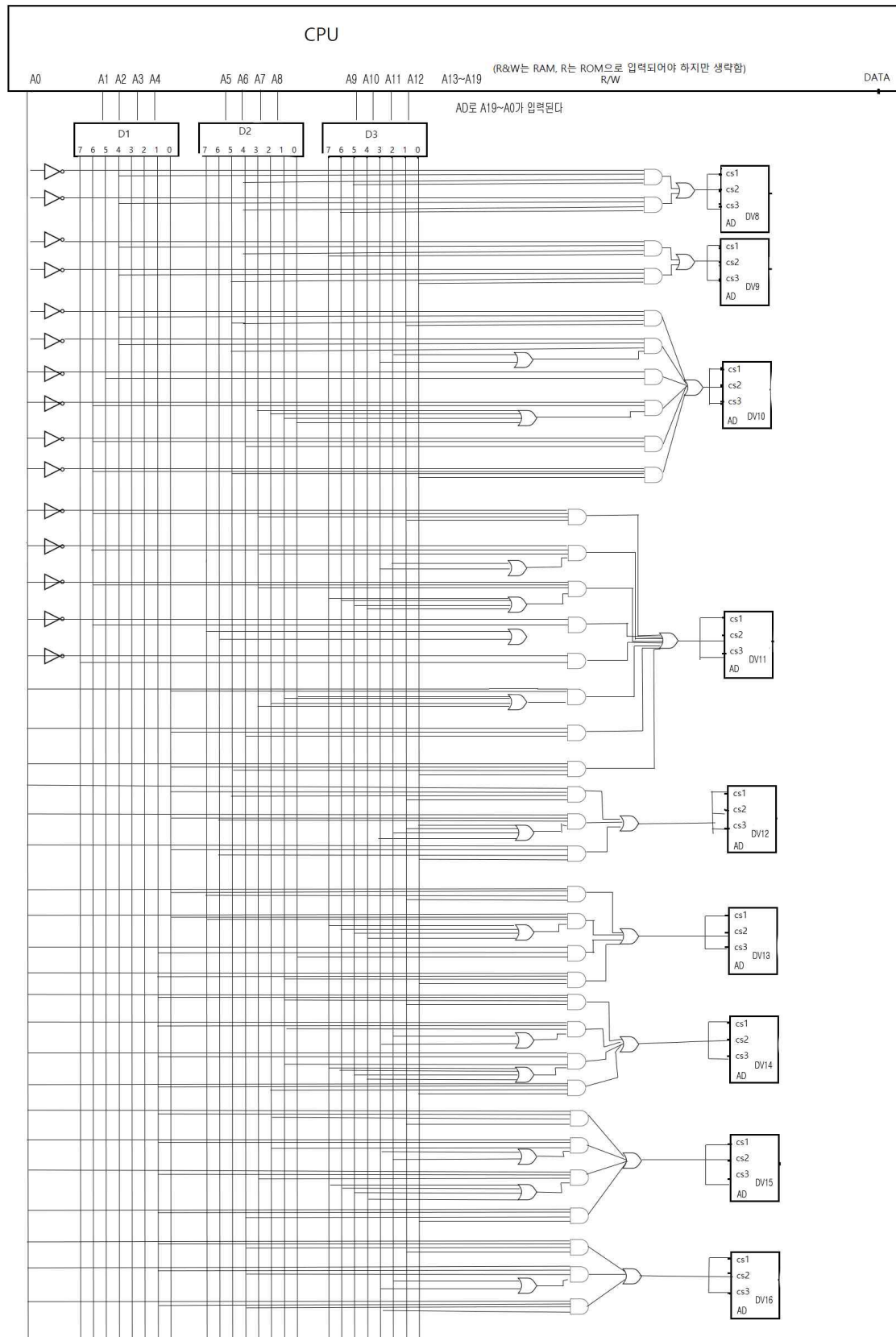
2. binary address map & Decoding

	A0~A3	A4~A7	A8~A11	A12~A15	A16~A19	A0& A1~A3(D1)	A4~A6(D2)	A7~A9(D3)
						,는 or임		
DV17	0000	0000	0000	0000	0000			
	0000	0000	1111	1111	1111			
	0000	0000	xxxx	xxxx	xxxx	0&0	0	0,1,2,3
DV18	0000	0001	0000	0000	0000			
	0000	0001	0111	1111	1111			
	0000	0001	0xxx	xxxx	xxxx	0&0	0	4,5
DV19	0000	0001	1000	0000	0000			
	0000	0001	1111	1111	1111			
	0000	0001	1xxx	xxxx	xxxx	0&0	0	6, 7
DV20	0000	0010	0000	0000	0000			
	0000	1001	1111	1111	1111			
	0000	001x	xxxx	xxxx	xxxx	0&0	1	-
	0000	01xx	xxxx	xxxx	xxxx	0&0	2,3	-
	0000	100x	xxxx	xxxx	xxxx	0&0	4	-
DV21	0000	1010	0000	0000	0000			
	0000	1101	1111	1111	1111			
	0000	101x	xxxx	xxxx	xxxx	0&0	5	-
	0000	110x	xxxx	xxxx	xxxx	0&0	6	-
DV22	0000	1110	0000	0000	0000			
	0000	1110	1111	1111	1111			
	0000	1110	xxxx	xxxx	xxxx	0&0	7	0,1,2,3
DV23	0000	1111	0000	0000	0000			
	0000	1111	0011	1111	1111			
	0000	1111	00xx	xxxx	xxxx	0&0	7	4
DV0	0000	1111	0100	0000	0000			
	0001	1111	0011	1111	1111			
	0000	1111	01xx	xxxx	xxxx	0&0	7	5
	0000	1111	1xxx	xxxx	xxxx	0&0	7	6,7
	0001	1111	00xx	xxxx	xxxx	0&1		
DV1	0001	1111	0100	0000	0000			
	0010	1111	0011	1111	1111			
	0001	1111	01xx	xxxx	xxxx	0&1	7	5
	0001	1111	1xxx	xxxx	xxxx	0&1	7	6,7
	0010	1111	00xx	xxxx	xxxx	0&2	7	4
DV2	0010	1111	0100	0000	0000			
	0011	0111	0011	1111	1111			
	0010	1111	01xx	xxxx	xxxx	0&2	7	5
	0010	1111	1xxx	xxxx	xxxx	0&2	7	6,7
	0011	0111	00xx	xxxx	xxxx	0&3	3	4
DV3	0011	0111	0100	0000	0000			
	0011	1111	0011	1111	1111			
	0011	0111	01xx	xxxx	xxxx	0&3	3	5

	0011	0111	1xxx	xxxx	xxxx	0&3	3	6,7
	0011	1111	00xx	xxxx	xxxx	0&3	7	4
DV4	0011	1111	0100	0000	0000			
	0100	0011	0011	1111	1111			
	0011	1111	01xx	xxxx	xxxx	0&3	7	5
	0011	1111	1xxx	xxxx	xxxx	0&3	7	6,7
	0100	0011	00xx	xxxx	xxxx	0&4	1	4
DV5	0100	0011	0100	0000	0000			
	0100	0111	0011	1111	1111			
	0100	0011	01xx	xxxx	xxxx	0&4	1	5
	0100	0011	1xxx	xxxx	xxxx	0&4	1	6,7
	0100	010x	xxxx	xxxx	xxxx	0&4	2	
	0100	0110	xxxx	xxxx	xxxx	0&4	3	0,1,2,3
	0100	0111	00xx	xxxx	xxxx	0&4	3	4
DV6	0100	0111	0100	0000	0000			
	0100	1000	0011	1111	1111			
	0100	0111	01xx	xxxx	xxxx	0&4	3	5
	0100	0111	1xxx	xxxx	xxxx	0&4	3	6,7
	0100	1000	00xx	xxxx	xxxx	0&4	4	0
DV7	0100	1000	0100	0000	0000			
	0100	1001	0011	1111	1111			
	0100	1000	01xx	xxxx	xxxx	0&4	4	1
	0100	1000	1xxx	xxxx	xxxx	0&4	4	2,3
	0100	1001	00xx	xxxx	xxxx	0&4	4	4
DV8	0100	1001	0100	0000	0000			
	0100	1001	1011	1111	1111			
	0100	1001	01xx	xxxx	xxxx	0&4	4	5
	0100	1001	10xx	xxxx	xxxx	0&4	4	6
DV9	0100	1001	1100	0000	0000			
	0100	1010	0011	1111	1111			
	0100	1001	11xx	xxxx	xxxx	0&4	4	7
	0100	1010	00xx	xxxx	xxxx	0&4	5	0
DV10	0100	1010	0100	0000	0000			
	0110	1010	0011	1111	1111			
	0100	1010	01xx	xxxx	xxxx	0&4	5	1
	0100	1010	1xxx	xxxx	xxxx	0&4	5	2,3
	0101	xxxx	xxxx	xxxx	xxxx	0&5	-	-
	0110	0xxx	xxxx	xxxx	xxxx	0&6	0,1,2,3	-
	0110	100x	xxxx	xxxx	xxxx	0&6	4	-
	0110	1010	00xx	xxxx	xxxx	0&6	5	0
DV11	0110	1010	0100	0000	0000			
	1000	1010	0011	1111	1111			
	0110	1010	01xx	xxxx	xxxx	0&6	3	1
	0110	1010	1xxx	xxxx	xxxx	0&6	3	2,3
	0110	1011	xxxx	xxxx	xxxx	0&6	3	4,5,6,7
	0110	11xx	xxxx	xxxx	xxxx	0&6	6,7	-
	0111	xxxx	xxxx	xxxx	xxxx	0&7	-	-

	1000	0xxx	xxxx	xxxx	xxxx	1&0	0,1,2,3	-
	1000	100x	xxxx	xxxx	xxxx	1&0	4	-
	1000	1010	00xx	xxxx	xxxx	1&0	5	0
DV12	1000	1010	0100	0000	0000			
	1000	1110	0011	1111	1111			
	1000	1010	01xx	xxxx	xxxx	1&0	5	1
	1000	1100	xxxx	xxxx	xxxx	1&0	6	0,1,2,3
	1000	1110	00xx	xxxx	xxxx	1&0	6	0
DV13	1000	1110	0100	0000	0000			
	1001	0010	0011	1111	1111			
	1000	1110	01xx	xxxx	xxxx	1&0	7	1
	1000	1111	xxxx	xxxx	xxxx	1&0	7	4,5,6,7
	1001	000x	xxxx	xxxx	xxxx	1&1	0	-
	1001	0010	00xx	xxxx	xxxx	1&1	1	0
DV14	1001	0010	0100	0000	0000			
	1001	0100	0011	1111	1111			
	1001	0010	01xx	xxxx	xxxx	1&1	1	1
	1001	0010	1xxx	xxxx	xxxx	1&1	1	2,3
	1001	0011	xxxx	xxxx	xxxx	1&1	1	4,5,6,7
	1001	0100	00xx	xxxx	xxxx	1&1	2	0
DV15	1001	0100	0100	0000	0000			
	1001	0110	0011	1111	1111			
	1001	0100	01xx	xxxx	xxxx	1&1	2	1
	1001	0100	1xxx	xxxx	xxxx	1&1	2	2,3
	1001	0101	xxxx	xxxx	xxxx	1&1	3	4,5,6,7
	1001	0110	00xx	xxxx	xxxx	1&1	4	0
DV16	1001	0110	0100	0000	0000			
	1001	0111	0011	1111	1111			
	1001	0110	01xx	xxxx	xxxx	1&1	4	1
	1001	0110	1xxx	xxxx	xxxx	1&1	4	2,3
	1001	0111	00xx	xxxx	xxxx	1&1	4	4





MSB인 A0와, {(A1,A2,A3), (A4,A5,A6), (A7,A8,A9)}를 입력으로 하는 3x8 decoder 3개를 이용하여 CPU와 메모리를 연결해주었다. 교과서에서 배운 것과 같이 CS를 담당하는 Address bit를 따로 사용할 수 있으면 편리하지만, 주어진 bit 개수와 메모리 주소 할당을 위한 bits 수가 같기 때문에 메모리에 할당된 주소 내에서 각 메모리를 구분해주는 방법을 사용하였다. 구분된 방식에 따라서, 1) D1&A0를 CS1에, D2를 CS2, D3를 CS3에 INPUT으로 사용하기도 하고, 2) 여러 가지 조합으로 구분된 경우에는 위와 마찬가지로의 방식을 사용하나 각각의 경우를 OR Gate에 input으로 넣은 후 output을 CS1~CS3에 입력시켜 마무리 지어줬다.

그림에도 적어놓았지만, RAM에는 Read와 Write, ROM에는 Read를 따로 입력해야 하지만 회로가 너무 복잡해지는 관계로 생략하였다.