Bit-Slice 형 CPU 의 Next Address 제어부와 Branch 제어부의 설계

○최 성훈 류 종필 정 호선 이 우일 *곽 명신 *유 영욱

경북대학교 전자공학과 , *한국전자통신연구소

Design of Next Address Control Unit and Branch Control Unit of Bit-Slice Type CPU

SUNG HOON CHOI , JONGPIL RYOO, HO SUN CHUNG
WU IL LEE , MUNG SIN GUAC , YOUNG UG YU

Dept. of Electronics, kyungpook national University.
*Electronics & Telecommunications Research Institute.

ABstract

The major objective of this paper is the design of control unit based on the bit slice technique.
The branch control unit is device that provides 16-way branch when used in conjuction with the Microprogram Sequencer.
The Next address control unit is designed specifically for next address control of the Microprogram Sequencer.

1. 서 론

CPU 는 컴퓨터의 가장 중심을 이루는 부분으로서 제어 부분과 자료 처리 장치로 되어 있다. MOS 기술로 제작되는 마이크로 프로세서는 대부분이 한 소자(one chip) 속에 포함되므로 집적도가 높아지는 장점이 있다. 그러나 MOS 마이크로 프로세서는 CPU의 구조, word의 길이, 명령어 조합 등이 미리 정해져 있으며 변경할수가 없다. 그리고 바이폴라 비트 슬라이스 마이크로 프로세서(1-3)는 CPU의 구조가 근본적으로 달라서 제어부와 자료 처리부가 각각 분리된 칩으로 되어 있다. 여러 개의 자료 처리부를 병렬 연결하면 용도에 따라 word의 길이를 확장(4)할수 있으며 마이크로 프로그래밍 방식에 의해 시스템의 명령어 조합을 정의할수 있다. 고속 마이크로 프로세서를 설계할 경우에는 바이폴라 기술을 사용하지만 낮은 집적도와 소비 전력의 증가를 초래한다. 그러나 CMOS standard cell을 이용하여 설계하면 비교적 바이폴라 기술에 비해 칩 면적을 줄일수 있고 소비 전력을 감소시킬수 있는 장점이 있다(5,6).

본 논문에서는 비트 슬라이스 마이크로 프로세서의 제어부인 Next address 제어부와 Branch 제어부를 설계하였다. 설계한 회로는 CMOS standard cell을 이용하여 자동 설계할수 있게 최대 게이트 입력의 수를 제한하였다.

2. Next address 제어부와 Branch 제어부의 구조

비트 슬라이스형 마이크로 프로세서의 제어부는 마이크로 프로그램 시이퀀서, 마이크로 프로그램 메모리, 마이크로 프로그램 시이퀀서의 다음 어드레스 제어를 위한 Next address 제어부와 마이크로 프로그램 시이퀀서의 최대 branch 능력을 제공해주는 Branch 제어부 등이 있다. 그림1은 비트 슬라이스형 마이크로 프로세서의 구성도이다. 하나의 Next address 제어부와 3개의 마이크로 프로그램 시이퀀서를 사용하면 4k word의 마이크로 프로그램 메모리를 제어할수 있다. 설계된 Next addresss 제어부는 4개의 명령어 입력들(I0-I3)과 하나의 TEST 입력의 조합에 의해 8개의 Next address 제어 신호를 제공하며 16개의 명령어 셋트를 구성한다. Branch 제어부는 4개의 TEST 입력들(T0-T3)과 4개의 I 입력들 (I0-I3)의 조합에 의해 한번의 마이크로 프로그램 시행 사이클에서 2, 4, 8 또는 16-way branch를 가능하게 한다.

3. 논리 회로 설계

1)Next address 제어부

4비트 명령어 입력들(I0-I3)들은 16개의 명령어들을 발생시킨다. Next address 제어부의 명령어와 그 기능 설명은 표1과 같다. Next address 제어부의 출력은 8개의 제어 신호를 제공한다. s1, s0는 마이크로 프로그램 시이퀀서의 s0와 s1 입력을 구동시켜서 마이크로 프로그램 메모리에 대한 다음 어드레스의 source로서 직접 입력, 레지스터, 마이크로 프로그램 카운터, 스택이 선택되게 한다. $\overline{FE}$ 신호는 마이크로 프로그램 시이퀀서의 file enable 입력을 구동하기 위해서 사용되며 출력이 low 이면 스택의 동작이 일어난다.

PUP 신호는 마이크로 프로그램 시이퀀서의 push/pop 입력을 구동시켜서 출력이 high 일때 file이 enable 되면 push가 일어나며 출력이 low이면 pop이 일어난다. PLE 출력은 branch address를 포함하는 파이프라인 레지스터의 3상태 출력을 제어하기 위해 사용된다.

MAP E 출력은 각 기계어에 대한 초기 starting address를 제공하기 위해 사용되는 mapping PROM의 3 상태 출력을 제어하기 위해 사용된다. CNT E는 up/down 카운터의 enable 입력을 구동하고 CNT LOAD는 카운터의 병렬 로드 입력을 구동하기 위해 사용된다.TEST 입력은 low일때 명령어의 테스트가 실패하고 high일때 명령어의 테스트가 성공한다.
그림2는 Next address 제어부의 논리 회로이다.

2) branch 제어부

설계된 Branch 제어부[7]는 한번의 마이크로 프로그램 시행 사이클에서 4 비트의 명령어의 조합에 의해 2,4,8 또는 16-way branch를 가능하게 한다. Branch 제어부의 Function table은 표2와 같다. 명령어 0는 disable 명령이고 1,2,4,8은 하나의 T입력을 테스트하고 2개의 워드 중 하나로 점프하게 한다. 7.11,13,14는 3개의 T 입력들을 테스트하고 8개의 워드 중 하나로 점프하게 한다. 15는 4개의 T입력들을 테스트하고 16개의 워드 중 하나로 점프하게 한다. 그림3은 Branch 제어부의 논리 회로이다. 회로의 복잡성 등을 고려하여 4개의 회로로 분리하여 설계하였다. 최대 게이트 입력의 수는 4로 하여 설계하였다.

4. 시뮬레이션 결과 및 고찰

설계된 논리 회로의 동작을 확인하기 위해서 논리 시뮬레이터인 EDAS_P(Electronic Design Automation System-Personal: 한국 전자 통신 연구소 제공)를 이용하였다. Next address 제어부와 Branch 제어부의 각 블록들을 논리 시뮬레이션하여 논리 동작이 정상적임을 확인하였다. 그림4는 Next address 제어부의 타이밍 다이아그램이다. TEST 입력이 low일때 명령어의 테스트가 실패하고 high 일때 명령어의 테스트가 성공한다는 것을 알수 있다. 그림5는 Branch address 제어부의 타이밍 다이아그램이다. 명령어 0은 disable 명령이고 명령어 1,2,4,8 일때 하나의 T 입력을 테스트하고 2개의 word 중 하나로 점프할수 있게 OR0를 구동시킨다는 것을 알수 있다. 명령어 3,5,6,9,10,12 일때는 2개 의 T 입력을 테스트하고 4개의 word 중 하나로 점프할수 있게 OR0,OR1을 구동시킨다. 명령어 7,11,13,14는 3개의 T 입력들을 테스트하고 8개의 word 중 하나로 점프할수있게 OR0,OR1,OR2를 구동시킨다. 명령어 15는 4개의 T-입력을 테스트하고 16개의 word중 하나로 점프할수 있게 OR0,OR1, OR2,OR3를 구동시킨다.

5. 결 론

마이크로 프로그램 시이퀀서와 관련되어 사용되는 Next address 제어부와 Branch 제어부를 CMOS standard cell을 이용하여 자동 설계할수 있게 논리회로를 설계하였다. 그리고 게이트에서의 최대 허용 입력 수를 4개로 제한하였다. Next address 제어부는 16개의 명령어를 제공하므로서 표2와 같은 기능에 따라 마이크로 프로그램 시이퀀서의 다음 어드레스를 제어할수 있게 하였다. Branch 제어부는 임의의 수의 T 입력을 테스트하여 최대 16-way branch 까지 가능하게 하였다. 실제 동작은 논리 시뮬레이션을 통하여 알아 보았다. 지금까지 설계된 회로와 자료 처리 장치 등을 이와 같이 설계하여 cell library 형식으로 구성한다면 새로운 컴퓨터 시스템의 개발이 가능하리라 본다.

6. 참고 문헌

1. John Mick and Jim Brick ,Bit slice Microprocessor Design,New York : McGraw-Hill,1980.
2. Advanced Micro devices,Bipolar Microprocessor Logic and Interface Data Book ,California: Advanced Micro Devices,1981.
3. Glenford J.Myers Digital System Design with LSI Bit-Slice _Logic : John Wiley & Sons,1980.
4. Donnamaie E. White,Bit-Slice Design: Controllers and ALUs, New York : Garland STPM Press. 1981.
5. Neil Weste,Principles for CMOS VLSI Design: Addison-Wesley,1985.
6. Design and Analysis of VLSI circuits : Addison-Wesley,1985
7. 이 재경 "Bit-Slice 형 마이크로프로세서 설계", 추계 종합 학술 대회 논문집,Vol.9,No.2,1986.

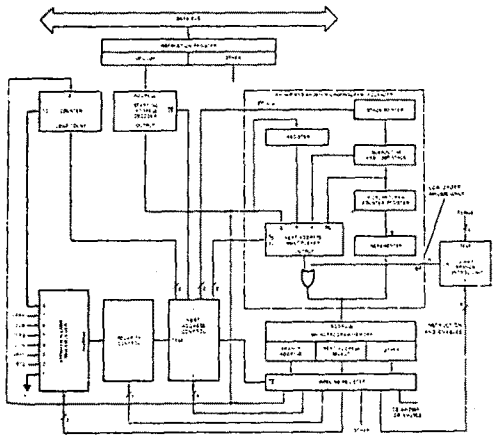


그림1. 비트 슬라이스형 마이크로 프로세서의 구성도

	BRANCH ON	I_3	I_2	I_1	I_0	OR_3	OR_2	OR_1	OR_0
NONE	NONE	L	L	L	L	L	L	L	L
Two-Way Branches	T_0	L	L	L	H	L	L	L	T_0
	T_1	L	L	H	L	L	L	L	T_1
	T_2	L	H	L	L	L	L	L	T_2
	T_3	H	L	L	L	L	L	L	T_3
Four-Way Branches	T_1 & T_0	L	L	H	H	L	L	T_1	T_0
	T_2 & T_0	L	H	L	H	L	L	T_2	T_0
	T_3 & T_0	H	L	L	H	L	L	T_3	T_0
	T_2 & T_1	L	H	H	L	L	L	T_2	T_1
	T_3 & T_1	H	L	H	L	L	L	T_3	T_1
	T_3 & T_2	H	H	L	L	L	L	T_3	T_2
Eight-Way Branches	T_2, T_1, T_0	L	H	H	H	L	T_2	T_1	T_0
	T_3, T_1, T_0	H	L	H	H	L	T_3	T_1	T_0
	T_3, T_2, T_0	H	H	L	H	L	T_3	T_2	T_0
	T_3, T_2, T_1,	H	H	H	L	L	T_3	T_2	T_1
Sixteen-Way Branch	T_3, T_2, T_1, T_0	H	H	H	H	T_3	T_2	T_1	T_0

표2. Branch 제어부의 Function

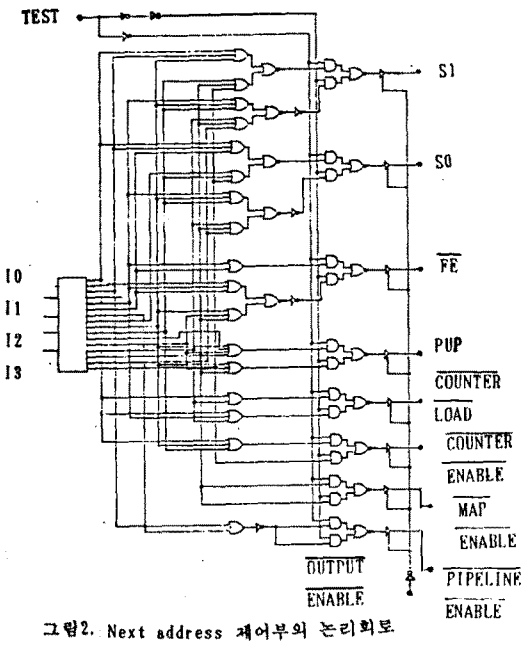
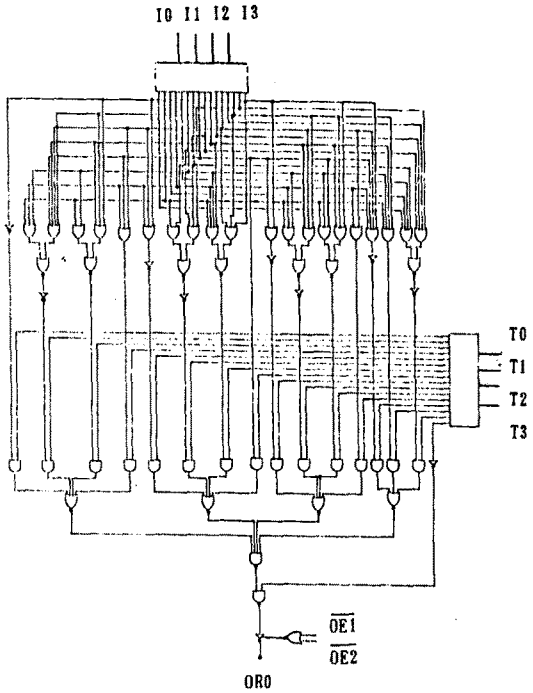


그림2. Next address 제어부의 논리회로

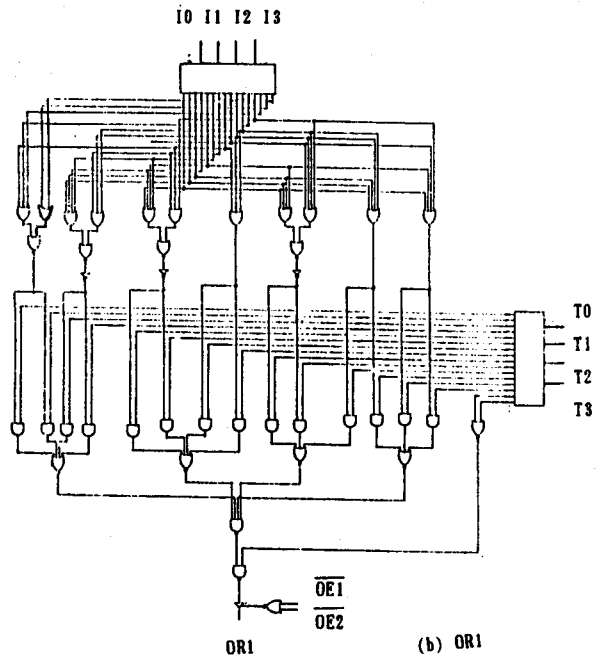
MNEMONIC	INSTRUCTION I_3 I_2 I_1 I_0	FUNCTION	TEST INPUT	NEXT ADDR SOURCE	FILE	COUNTER	MAP-E	PL-E
JZ	L L L L	JUMP ZERO	X	D	HOLD	L L*	H	L
CJS	L L L H	COND JSB PL	L	PC	HOLD	HOLD	H	L
			H	D	PUSH	HOLD	H	L
JMAP	L L H L	JUMP MAP	X	D	HOLD	HOLD	L	H
CJP	L L H H	COND JUMP PL	L	PC	HOLD	HOLD	H	L
			H	D	HOLD	HOLD	H	L
PUSH	L H L L	PUSH/COND LD CNTR	L	PC	PUSH	HOLD	H	L
			H	PC	PUSH	LOAD	H	L
JSRP	L H L H	COND JSB R/PL	L	R	PUSH	HOLD	H	L
			H	D	PUSH	HOLD	H	L
CJV	L H H L	COND JUMP VECTOR	L	PC	HOLD	HOLD	H	H
			H	D	HOLD	HOLD	H	H
JRP	L H H H	COND JUMP R/PL	L	R	HOLD	HOLD	H	L
			H	D	HOLD	HOLD	H	L
RPCT	H L L L	REPEAT LOOP, CNTR ≠ 0	L	F	HOLD	DEC	H	L
			H	PC	POP	HOLD	H	L
RCCT	H L L H	REPEAT PL, CNTR ≠ 0	L	D	HOLD	DEC	H	L
			H	PC	HOLD	HOLD	H	L
CRTN	H L H L	COND RTN	L	PC	HOLD	HOLD	H	L
			H	F	POP	HOLD	H	L
CJPP	H L H H	COND JUMP PL & POP	L	PC	HOLD	HOLD	H	L
			H	D	POP	HOLD	H	L
LDCT	H H L L	LOAD CNTR & CONTINUE	X	PC	HOLD	LOAD	H	L
LOOP	H H L H	TEST END LOOP	L	F	HOLD	HOLD	H	L
			H	PC	POP	HOLD	H	L
CONT	H H H L	CONTINUE	X	PC	HOLD	HOLD	H	L
JP	H H H H	JUMP PL	X	D	HOLD	HOLD	H	L

표1. Next address 제어부의 기능



(a) OR0

그림3. Branch 제어부의 논리 회로



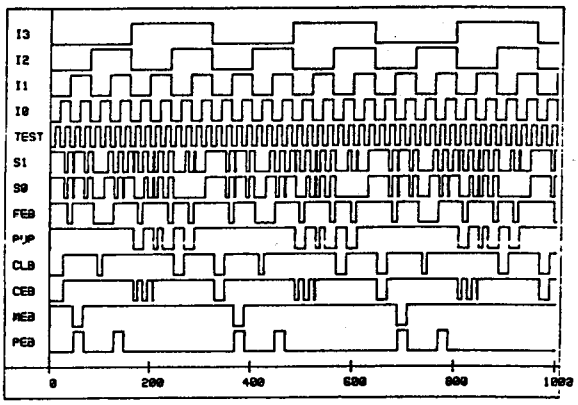


그림4. Next address 제어부의 타이밍 다이아그램

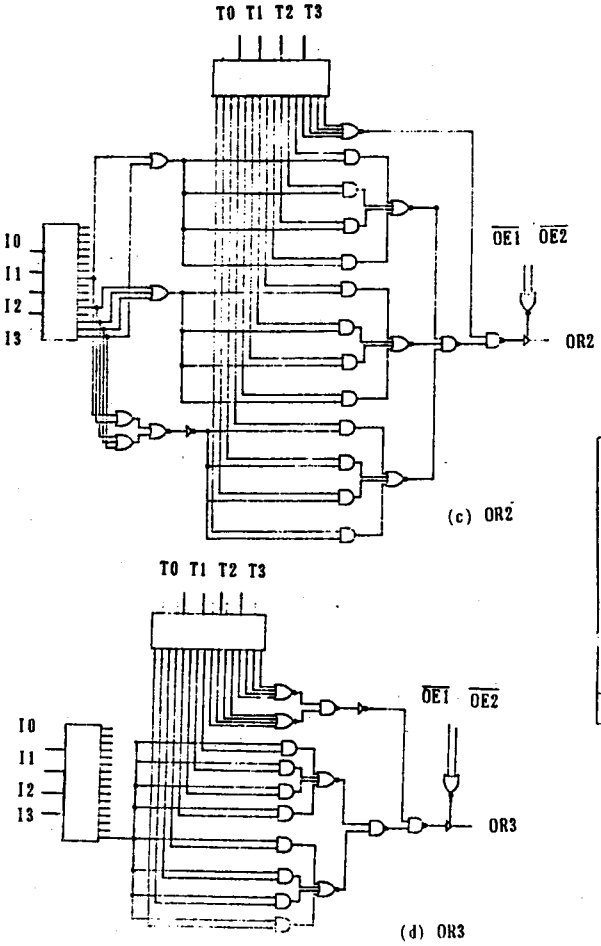
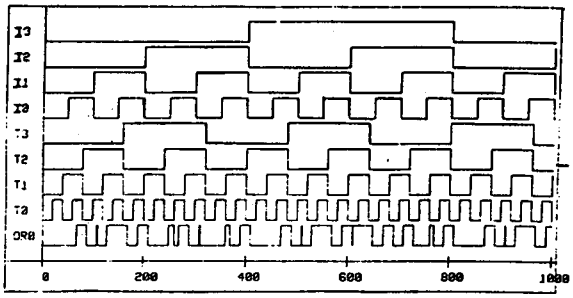
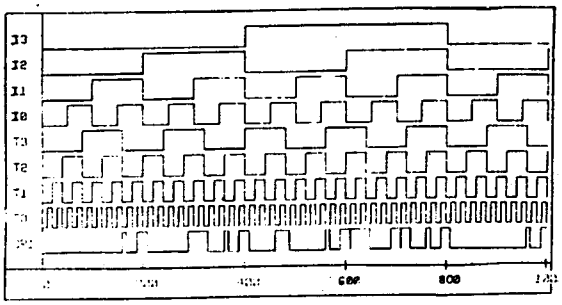


그림3. Branch 제어부의 논리 회로



(a) OR0



(b) OR1

그림5. Branch 어드레스 제어부의 타이밍 다이아그램

** 본 연구는 한국전자통신연구소의 위탁과제로 이루어진 것이며 연구소에 감사드립니다. **