


논리회로및설계 범용튜링머신

(8 States 4 Symbols)

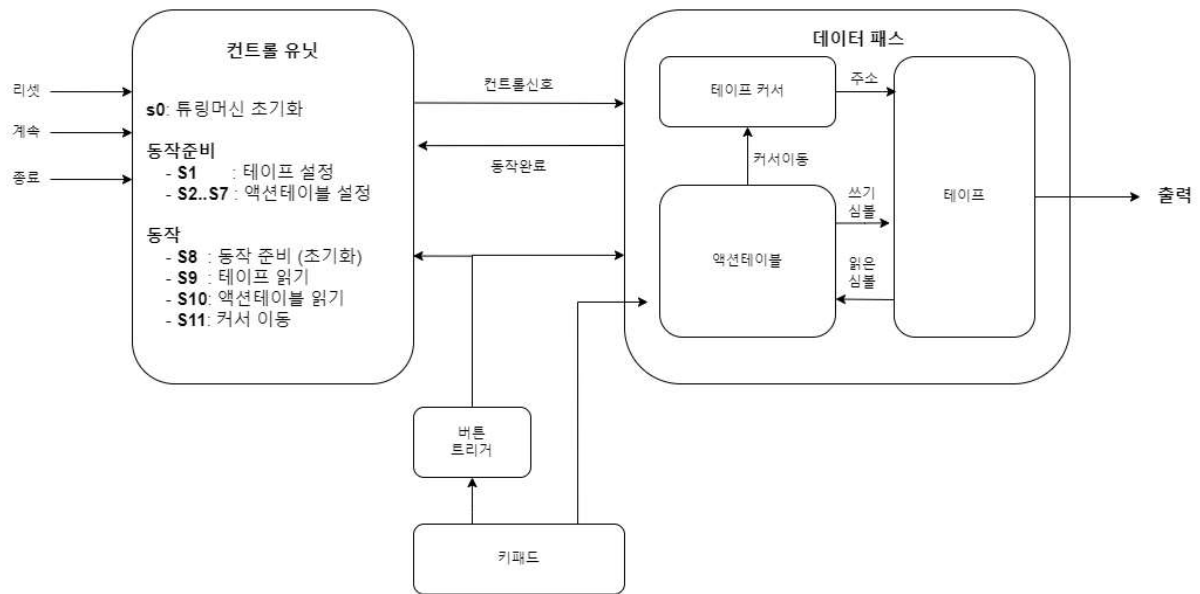


장원석(전기컴퓨터공학부)

윤 (정보컴퓨터공학부)

Nov 21, 2021

1. 아키텍처



2. 역할 분담

단계	항목	담당
설계	아키텍처 설계, 필요모듈 식별	장원석
구현	컨트롤 유닛	장원석
	테이프 헤드 (커서)	장원석
	액션테이블	장원석
	테이프 레지스터	윤선재
	입출력	윤선재
테스트	조립 및 연결	장원석
	디버깅	공동

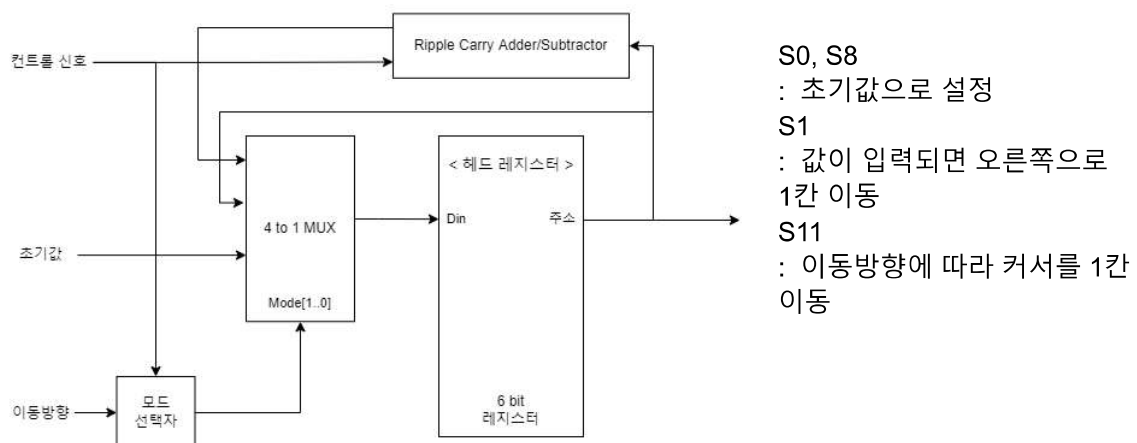
3. 모듈별 설명

1) 컨트롤유닛 (ControlUnit.bdf)

데이터패스의 각 모듈이 타이밍에 맞추어 동작하게끔 상태신호를 전달한다. 주로 레지스터에 대한 enabler 역할을 하며, 상태신호는 13 bit의 One-Hot 버스로 구성된다.

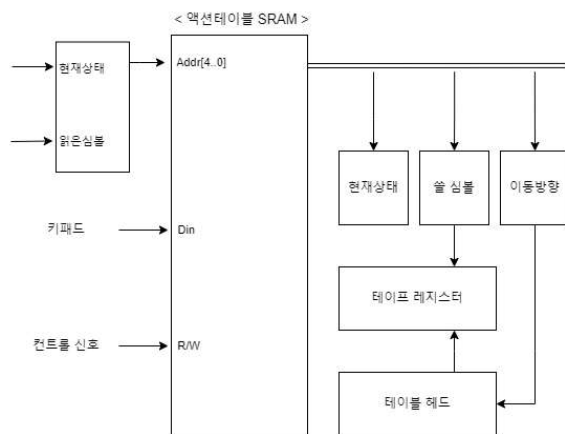
상태	역할
S0	데이터패스의 모든 모듈을 초기화한다.
S1	0~3의 숫자값을 받아 초기 테이프 심볼을 설정한다.
S2~S6	키패드로 규칙표를 입력받고 액션테이블에 기록한다. <ul style="list-style-type: none"> • S2: 현재 상태 (0~7) • S3: 읽을 테이프 심볼 (0~3) • S4: 다음 상태 (0~7) • S5: 쓸 테이프 심볼 (0~3) • S6: 이동할 방향 (0: 제자리, 1: 왼쪽, 2: 오른쪽)
S7	액션테이블 입력단계 종료여부 선택 (계속/종료 신호)
S8	튜링 머신 동작 이전에 커서 위치와 시작상태, 출력텍스트를 설정한다.
S9~S11	액션테이블을 바탕으로 튜링머신을 동작시킨다. <ul style="list-style-type: none"> • S9: 커서가 가리키는 테이프 심볼을 읽는다. • S10 <ul style="list-style-type: none"> ◦ 다음상태, 쓸심볼, 이동방향 읽기 (액션테이블) ◦ 쓸심볼을 테이프 심볼에 쓰기 • S11: 이동방향에 따라 커서를 이동시킨다.
S12	튜링머신이 종료된 상태. 현재상태와 읽은 심볼에 변동이 없으면 종료된다. (미구현)

2) 테이프 헤드 (Cursor.bdf)



테이프 헤드는 테이프 레지스터의 64칸 중 하나를 가리키는 주소 역할을 한다. 항상 1칸씩 왼쪽/오른쪽 이동하고 컨트롤 신호와 이동방향을 받아 그 동작을 결정한다.

3) 액션테이블 (TableRegister.bdf)



* 저장소: 32x8bit SRAM

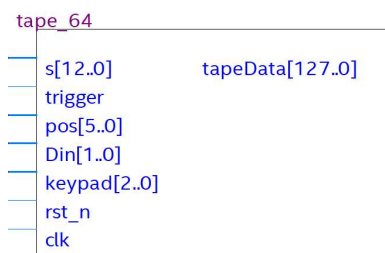
- 주소선: 현재상태 + 읽은심볼
=> 3bit + 2bit = 5bit
- 데이터: 다음상태, 쓸심볼, 이동방향
=> 3bit + 2bit + 2bit = 7bit (8bit)

* 컨트롤신호

- S4~S7: 데이터 입력받기
- S9: 현재상태 변경
- S10: 테이프 심볼 읽기

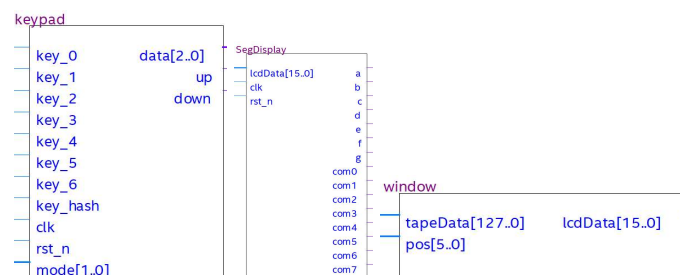
SRAM은 상기한 데이터를 저장하고, 데이터는 타이밍에 맞추어 각각 현재상태, 테이프, 테이블 헤드에 전달된다.

4) 테이프 레지스터 (tape_64.bdf)



테이프는 총 64개의 심볼 데이터를 담을 수 있으며 각 심볼은 2비트의 크기를 가지므로 테이프의 크기는 128비트이다. 테이프는 초기화 단계인 S1과 튜링머신이 동작 단계 중 테이프의 심볼을 수정하는 S10일 때, 수정하고자 하는 위치 pos와 수정하려는 Din을 입력하여 해당 위치의 심볼을 수정할 수 있다. tapeData는 현재 테이프의 데이터들을 128비트의 버스로 출력한다.

5) 입출력 (keypad.bdf, window.bdf, SegDisplay.bdf)

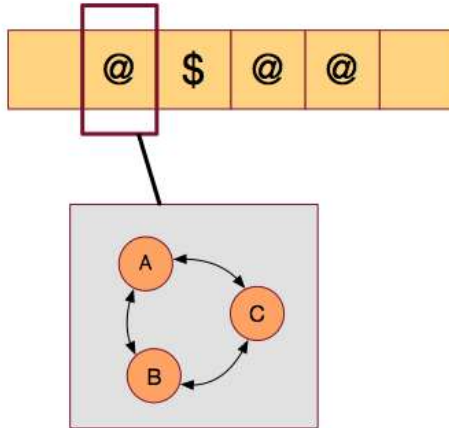


keypad는 0~6과 #의 총 8가지의 버튼들이 입력될 때, 버튼이 눌렸을 때(down), 버튼을 올라올 때(up), 그리고 눌린 버튼의 데이터(0~6, #는 7)를 3비트 버스로 나타낸 data[2..0]을 출력한다.

window는 128비트 크기의 tapeData와 헤드의 위치 pos를 입력받아 실제로 디스플레이 될 8개의 심볼, 총 16비트 크기의 데이터들을 출력한다.

SegDisplay는 window에서 디스플레이할 데이터 lcdData[15..0]을 받아 이를 실제로 출력하기 위한 com과 세그먼트 데이터 a~g를 출력한다.

4. Usecase



Current State	Read Symbol	Write Symbol	Next Position	Next State
A	@	@	->	A
A	\$	\$	->	B
B	@	@	<-	C
C	\$	@	->	C
C	@	\$	->	B

튜링머신은 연산 장치의 일종이고, 범용튜링머신은 규칙표(액션테이블)이 주어진다면 임의의 튜링머신을 흉내낼 수 있다. 이번 설계에서는 시간 상 8개 상태, 4개 심볼, 64칸 테이프를 한정했고, 이 제약 내의 모든 규칙표를 시뮬레이션할 수 있다. 위는 그중 '1' 심볼 사이의 '2' 심볼을 오른쪽 끝으로 옮기는 예시이다.

5. 구현 성과

동작에 필요한 모듈은 모두 구현했으나 기능 상 보완이 필요한 모듈이 여럿 존재한다. 다음은 위의 주요 모듈을 위한 서브모듈 목록과 개선이 필요한 부분들이다.

1) 서브 모듈

모듈명	설명	활용
Compare(n)bit	두 데이터가 동일한지 판단	IsStable.bdf (안정상태 판단)
Mux	2x1, 4x1, 8x1 멀티플렉서를 버스 폭에 맞추어 각각 구현	액션테이블, 테이프 헤드, 입출력
DeMux	1x2 디멀티플렉서를 버스 폭에 맞추어 각각 구현	SRAM 주소+R/W 선택기 등
OneHotEncoder OneHotDecoder	이진수 <-> OneHot 형식 변환	SRAM 주소+R/W 선택기, 출력 등
RippleCarry Adder/Subtractor	HalfAdder, FullAdder를 이용한 6bit 덧셈/뺄셈기	테이프 헤드의 커서 이동 커서 기준 양옆 3칸의 주소 계산
Up/Down Trigger	버튼이 눌렸을 때, 떼어졌을 때 1클럭을 발생시키는 트리거	컨트롤유닛, 데이터패스

2) 보완 필요

- 테이프 설정은 액션테이블 구성 이후가 낫다. (S1 상태 이동)
- 안정상태(튜링머신 종료, S12)를 판단해야 한다.
- 안정상태(S12)에서 커서를 자유롭게 이동할 수 있어야 한다.
- 액션테이블 구성 시 각 단계에 맞는 기호를 출력해야 한다.