HM_Vending

목 차

개 요

- 목 적 : Verilog로 설계된 Vending Machine
- 제작언어 : Verilog
- 사용 툴 : ModelSim, Xillinx

Spec & Function

- 총 4개의 상품 판매 가능한 제품
  - 400, 500, 600, 700원으로 진열

- 십진수 4자리 연산의 구현
  - 100원 - 1000원까지 주화 가능
  - 1000원 단위까지 7- Segment 표시

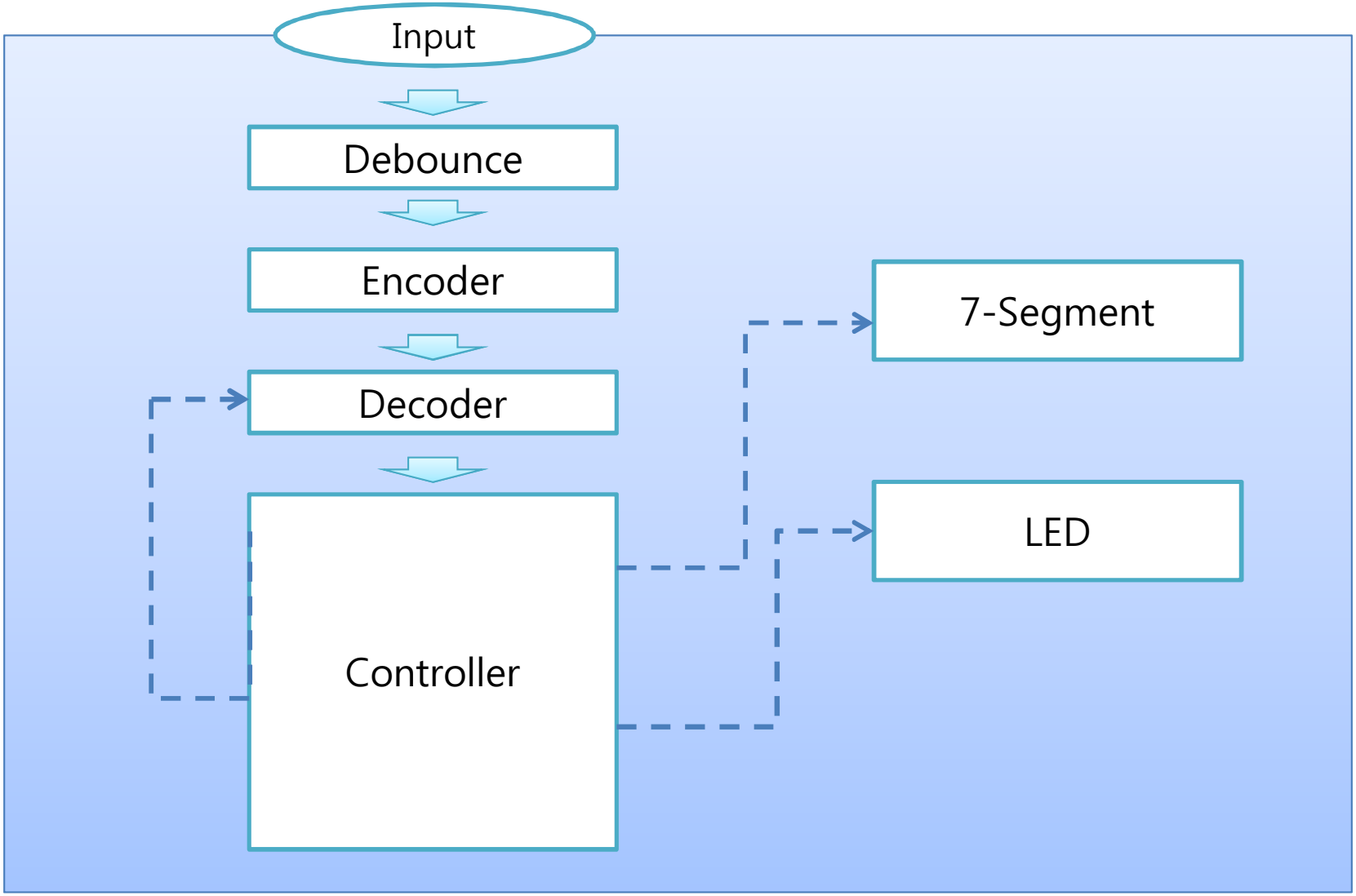
- 푸시 버튼 / 키 패드를 통한 입력

- LED / 7-segment를 통한 출력

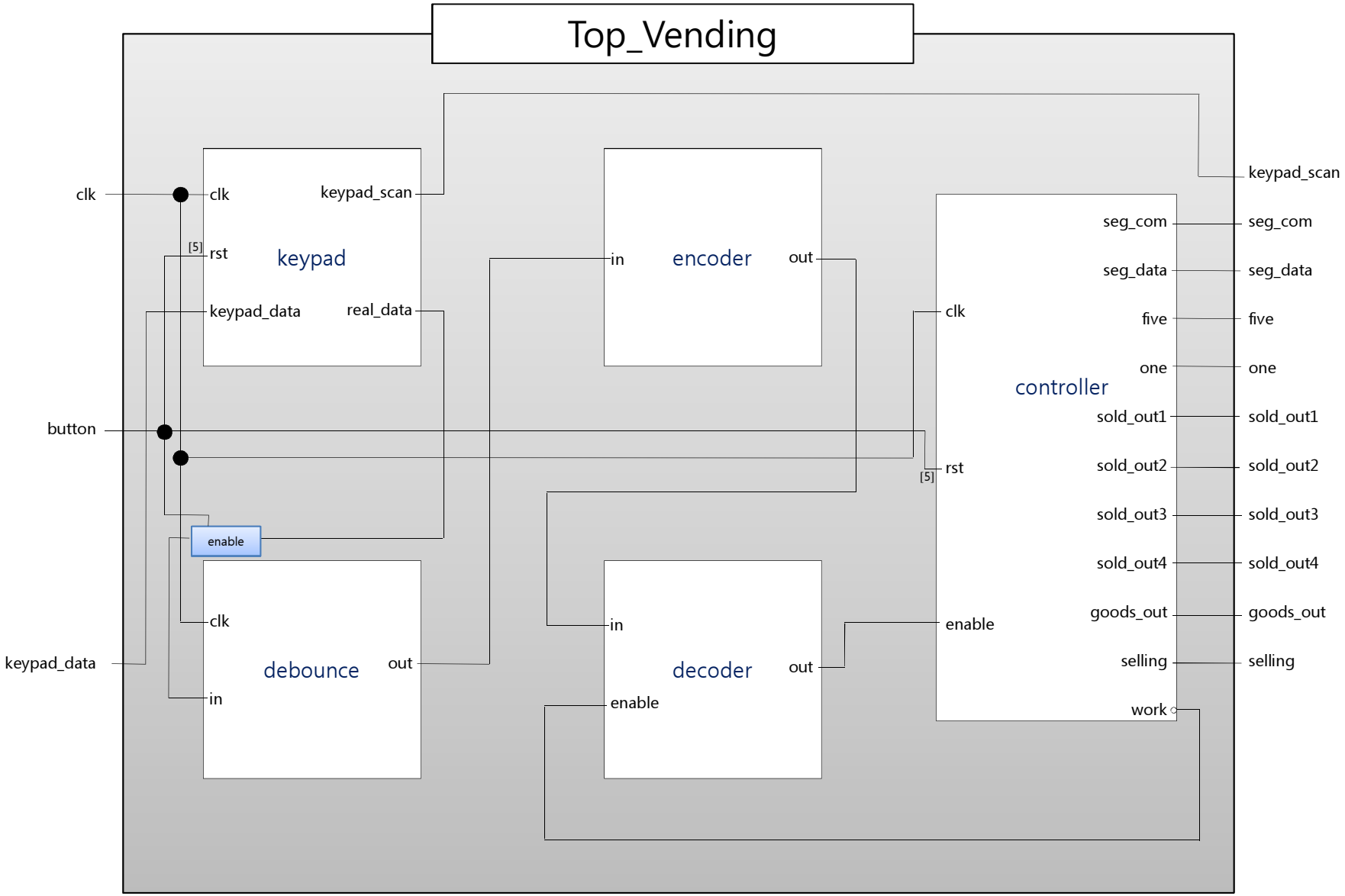
- 관리자 모드 상품 수량 입력

한백 Empos – II 보드

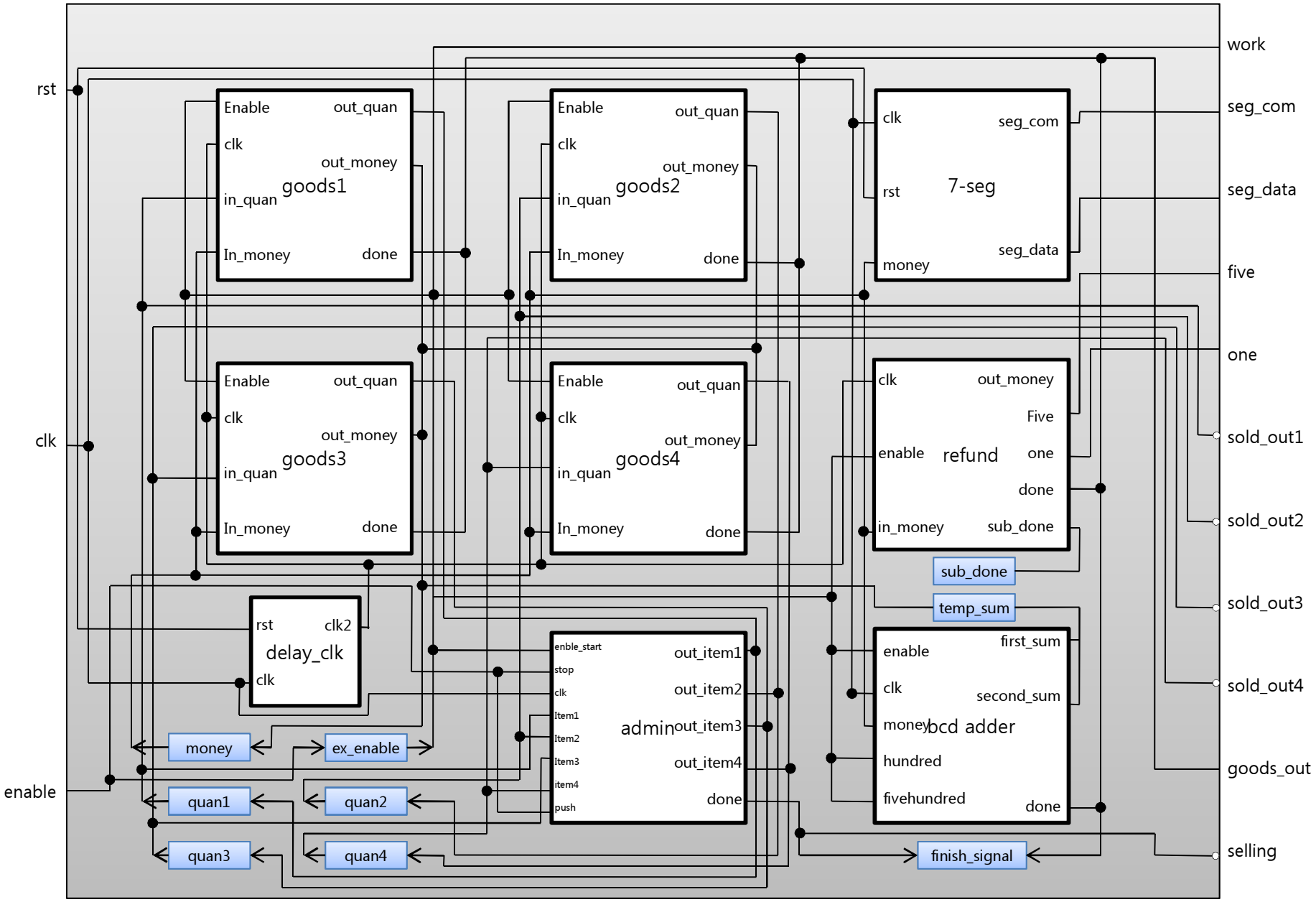
동작원리



전체 블럭도


Top_Vending
clk
button
keypad_data
clk
[5]
rst
keypad_data
keypad_scan
real_data
keypad
enable
clk
in
debounce
out
in
encoder
out
in
enable
decoder
out
clk
[5]
rst
enable
controller
seg_com
seg_data
five
one
sold_out1
sold_out2
sold_out3
sold_out4
goods_out
selling
work
keypad_scan
seg_com
seg_data
five
one
sold_out1
sold_out2
sold_out3
sold_out4
goods_out
selling

Controller 블럭도



컨 트 롤 러 설 명

컨트롤러 모듈 목적

- 총체적인 자판기를 통제, 관리
- 디코더에서 나온 신호의 기억 후 디코더 모듈 정지
- 돈과 재고량을 입력으로 저장, 다른 모듈에게 연결
- 돈을 항상 최신화, 재고량 매진 여부 판단

제 품 모 듈 목 적

- 5개의 제품 모듈로 구성
- 각 제품의 재고량과 값을 저장

모 듈 설 명

관리자 모듈 목 적

- 메뉴의 종류, 재고량 입력으로 받음
- 메뉴버튼에 따른 재고량을 하나씩 올림 ▶ Adder
- 완료 버튼을 누르면 컨트롤러에게 완료 신호 전달

반 환 모 듈 목 적

- 컨트롤러에게 현재돈을 입력으로 받음
- 현재돈을 0으로 초기화
- 결과를 7-Seg 모듈로 보낸 후 출력

모 듈 설 명

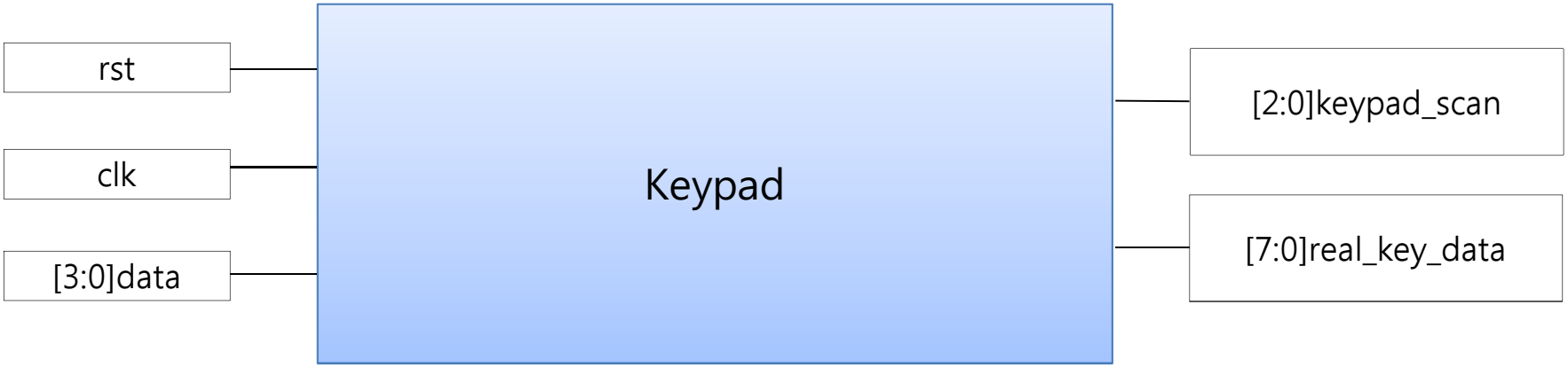
덧셈모듈 목적

- 현재 돈 값과 입력된 돈 값의 정보를 받음
- BCD Adder를 활용 값을 변환
- 변환된 값을 7-seg, 컨트롤러에게 전달

뺄셈모듈목적

- 컨트롤러에게 현재돈과 상품 가격을 입력으로 받음
- 현재돈과 상품 가격을 minus 후 값을 변환
- 결과를 7-seg, 컨트롤러에게 전달
- 상품의 재고량을 -1 한 후 전달

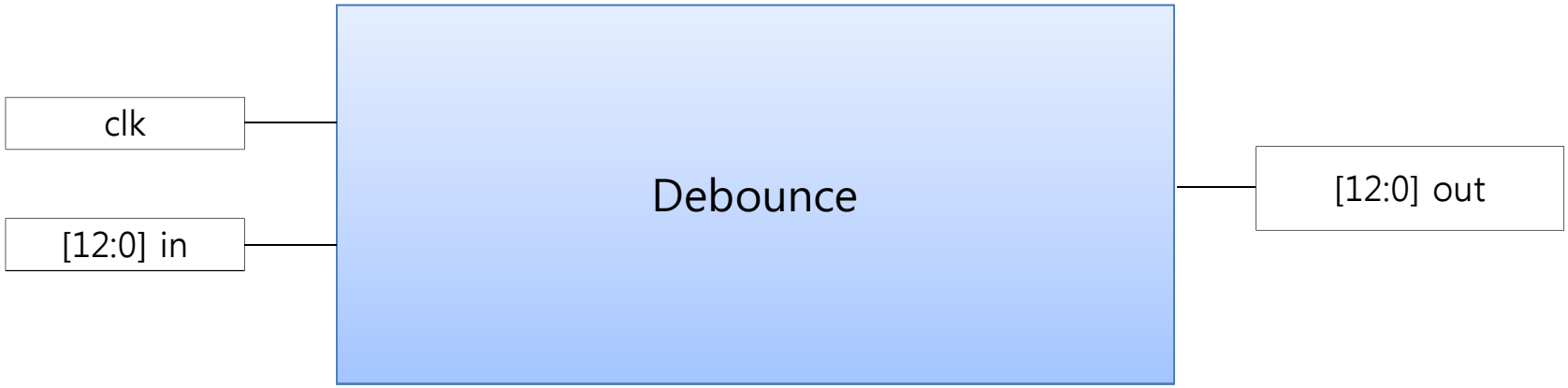
Module (Keypad)



Module (Keypad)

Pin Name	I/O	T/S	Description
rst	In	button[5] (Top)	초기화 버튼
clk	In	clk (Top)	clk
data	In	key_data (Top)	Button 입력
keypad_scan	Out	key_scan (Top)	Button 탐색
real_data	Out	enable (Top)	최종값 출력

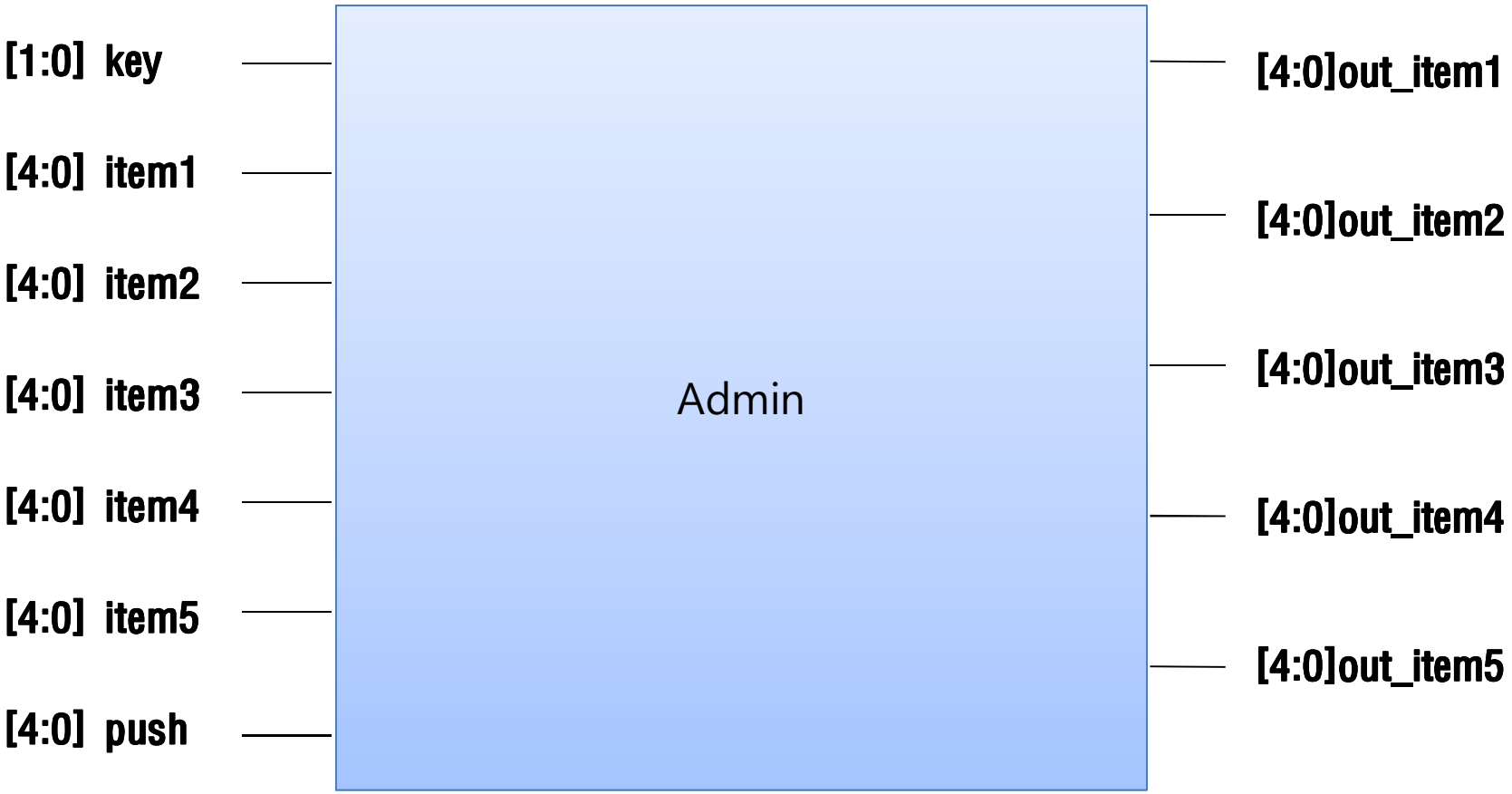
Module (Debounce)



Module (Debounce)

Pin Name	I/O	T/S	Description
clk	In	clk (Top)	clk
in	In	enable (Top)	입력 버튼
out	Out	in (encoder)	enable 신호

Module (admin)



Signal 구성도

Input

[1:0] key : Start and End Key
[4:0] item : 상품 모듈로 부터 item 수량 입력 (5)
[4:0] push : 상품의 수량을 +1

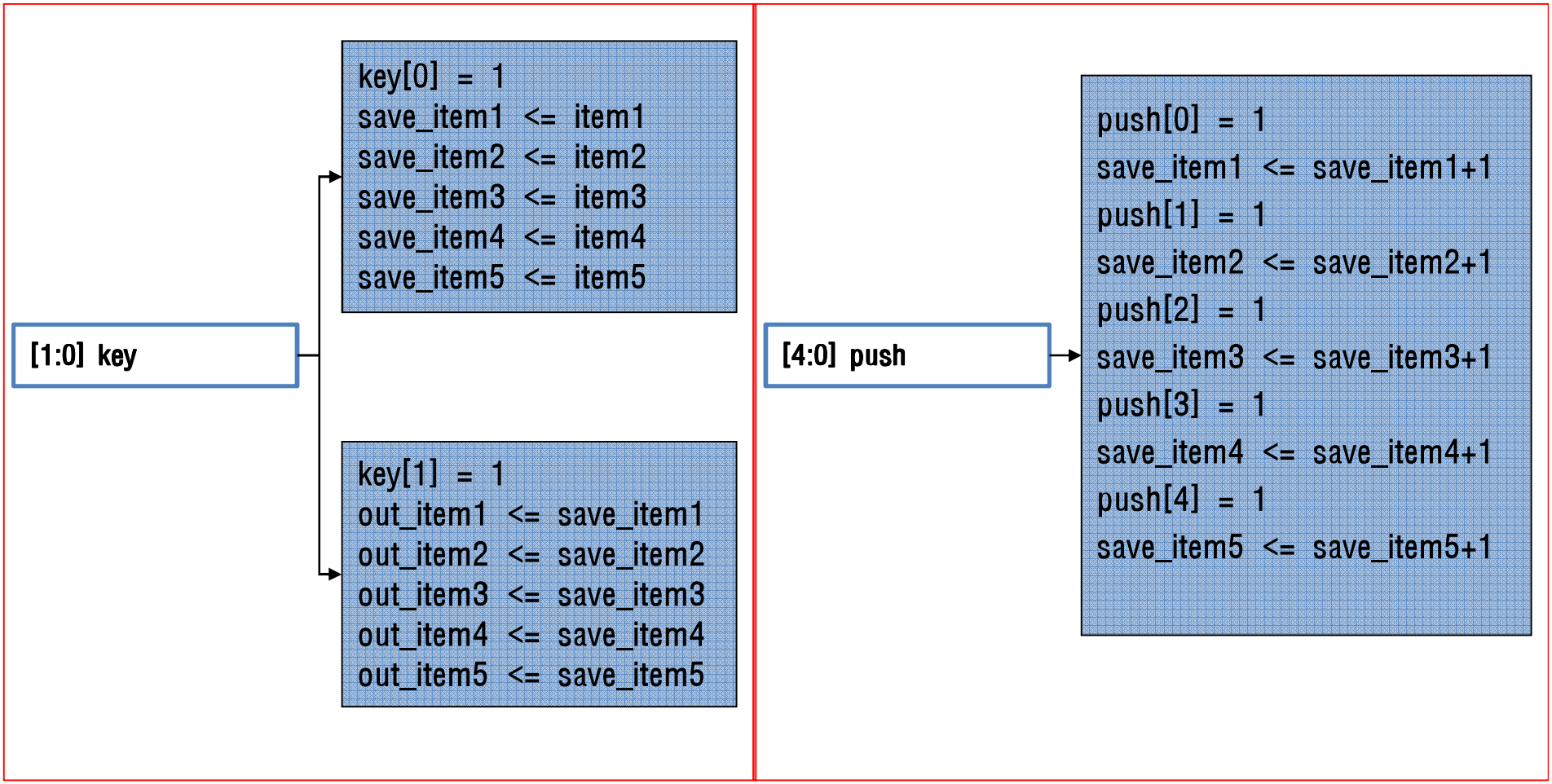
Output

[4:0] out_item : 계산된 item의 수량을 출력 (5)

Reg

[4:0] save_item : 계산중 item의 수량을 임시저장 (5)

모듈 알고리즘


[1:0] key
key[0] = 1
save_item1 <= item1
save_item2 <= item2
save_item3 <= item3
save_item4 <= item4
save_item5 <= item5
key[1] = 1
out_item1 <= save_item1
out_item2 <= save_item2
out_item3 <= save_item3
out_item4 <= save_item4
out_item5 <= save_item5
[4:0] push
push[0] = 1
save_item1 <= save_item1+1
push[1] = 1
save_item2 <= save_item2+1
push[2] = 1
save_item3 <= save_item3+1
push[3] = 1
save_item4 <= save_item4+1
push[4] = 1
save_item5 <= save_item5+1

관리자 모듈 소스

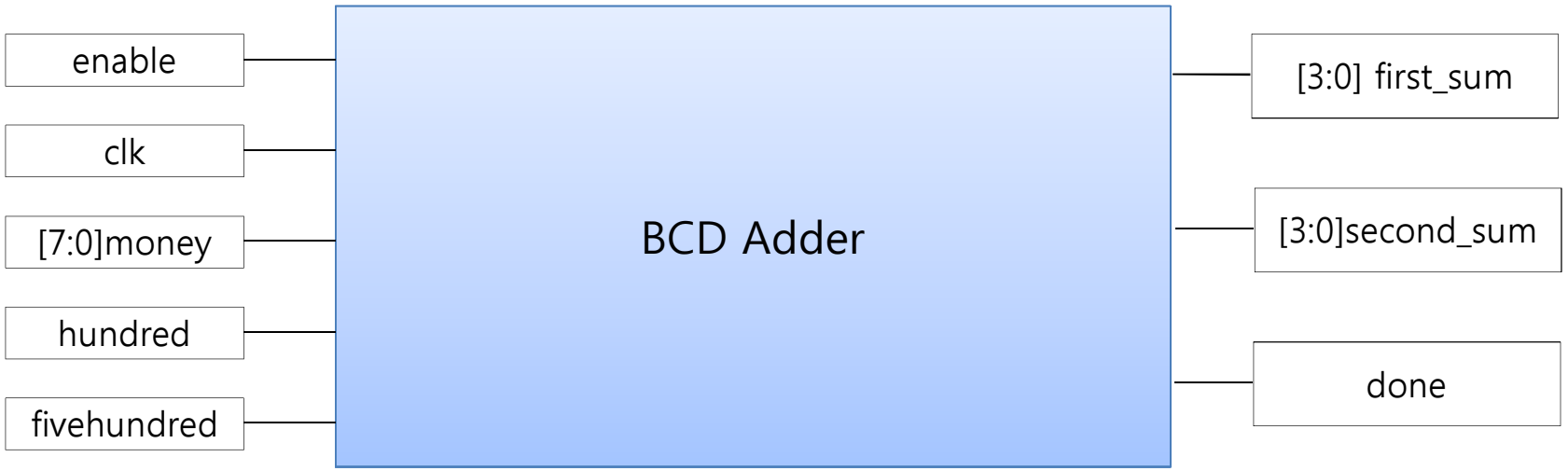
```
always @(key) begin
                    if(key[0] == 1)begin
                    save_item1 <= item1;
                    save_item2 <= item2;
                    save_item3 <= item3;
                    save_item4 <= item4;
                    save_item5 <= item5;
                    end
                    else if(key[1] == 1)begin
                    out_item1 <= save_item1;
                    out_item2 <= save_item2;
                    out_item3 <= save_item3;
                    out_item4 <= save_item4;
                    out_item5 <= save_item5;
          end

always @(push) begin
                    if(push[0] == 1) save_item1 <= save_item1 + 1;
                    else if(push[1] == 1) save_item2 <= save_item2 + 1;
                    else if(push[2] == 1) save_item3 <= save_item3 + 1;
                    else if(push[3] == 1) save_item4 <= save_item4 + 1;
                    else if(push[4] == 1) save_item5 <= save_item5 + 1;
                    end
```

reg ⇨ ⇦ input

output ⇨

Module (BCD Adder)



Module (BCD Adder)

Pin Name	I/O	T/S	Description
enable	In	enable (controller)	Enable 신호
clk	In	clk (Top)	CLK
[7:0]money	In	money (controller)	기존에 있는 돈
hundred	In	enable (controller)	백원 신호
fivehundred	In	enable (controller)	오백원 신호
[3:0]first_sum	Out	money (controller)	백단위 출력
[3:0]second_sum	Out	money (controller)	천단위 출력
done	Out	selling (controller)	End 신호

BCD Adder

BCD Adder의 의미

- Binary Coded Decimal Adder (이진화 십진수 덧셈기)

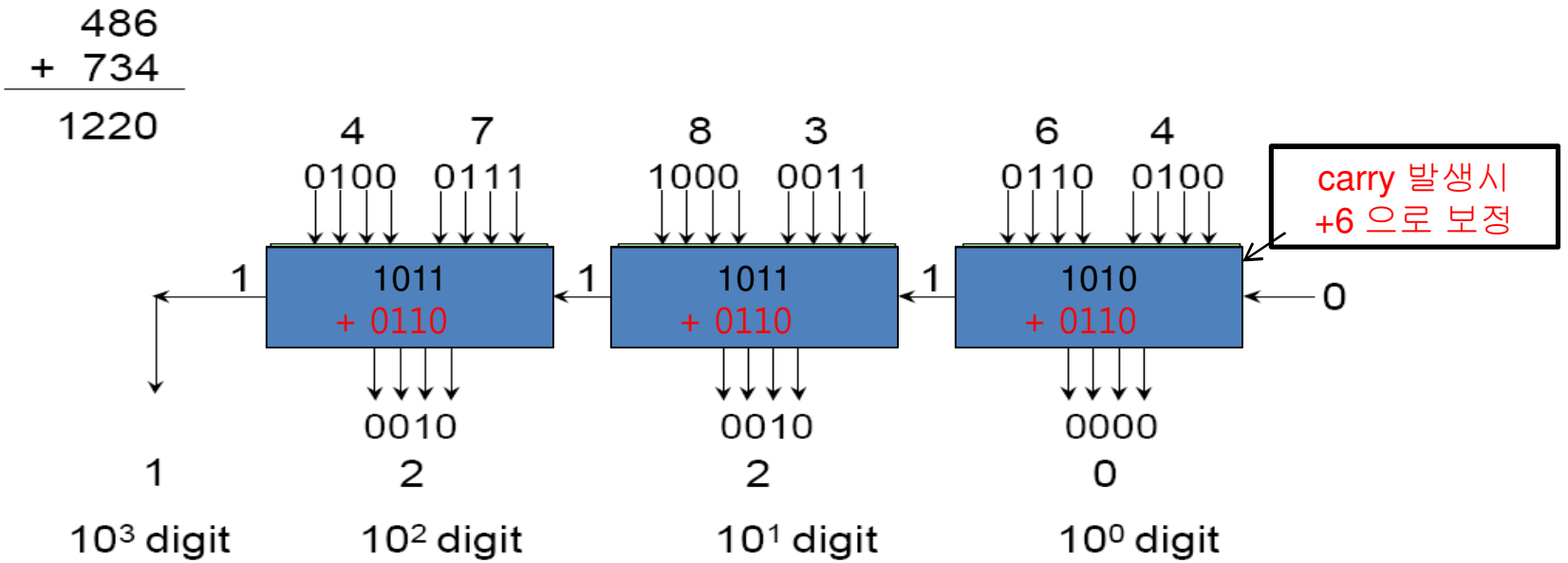
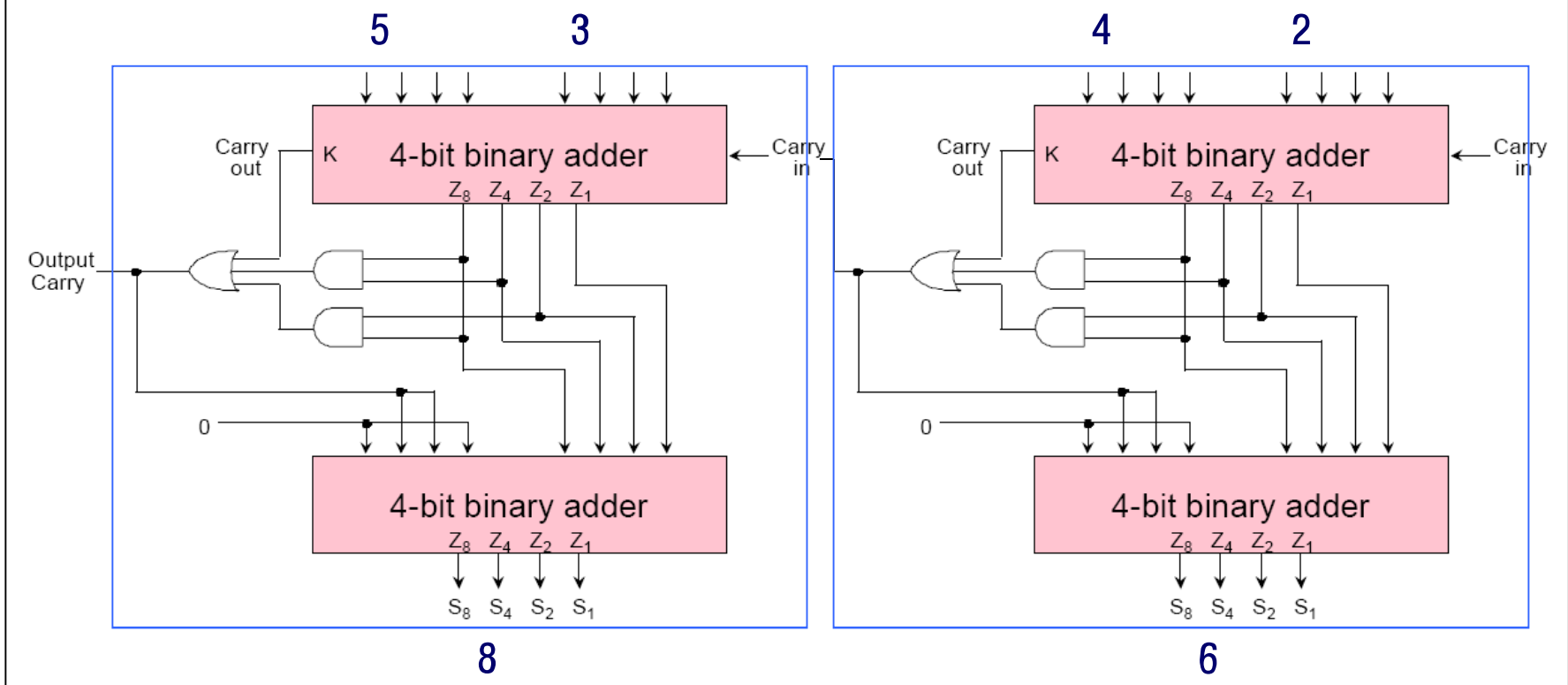


그림 출처 - 이재흥 교수님 수업 자료 -

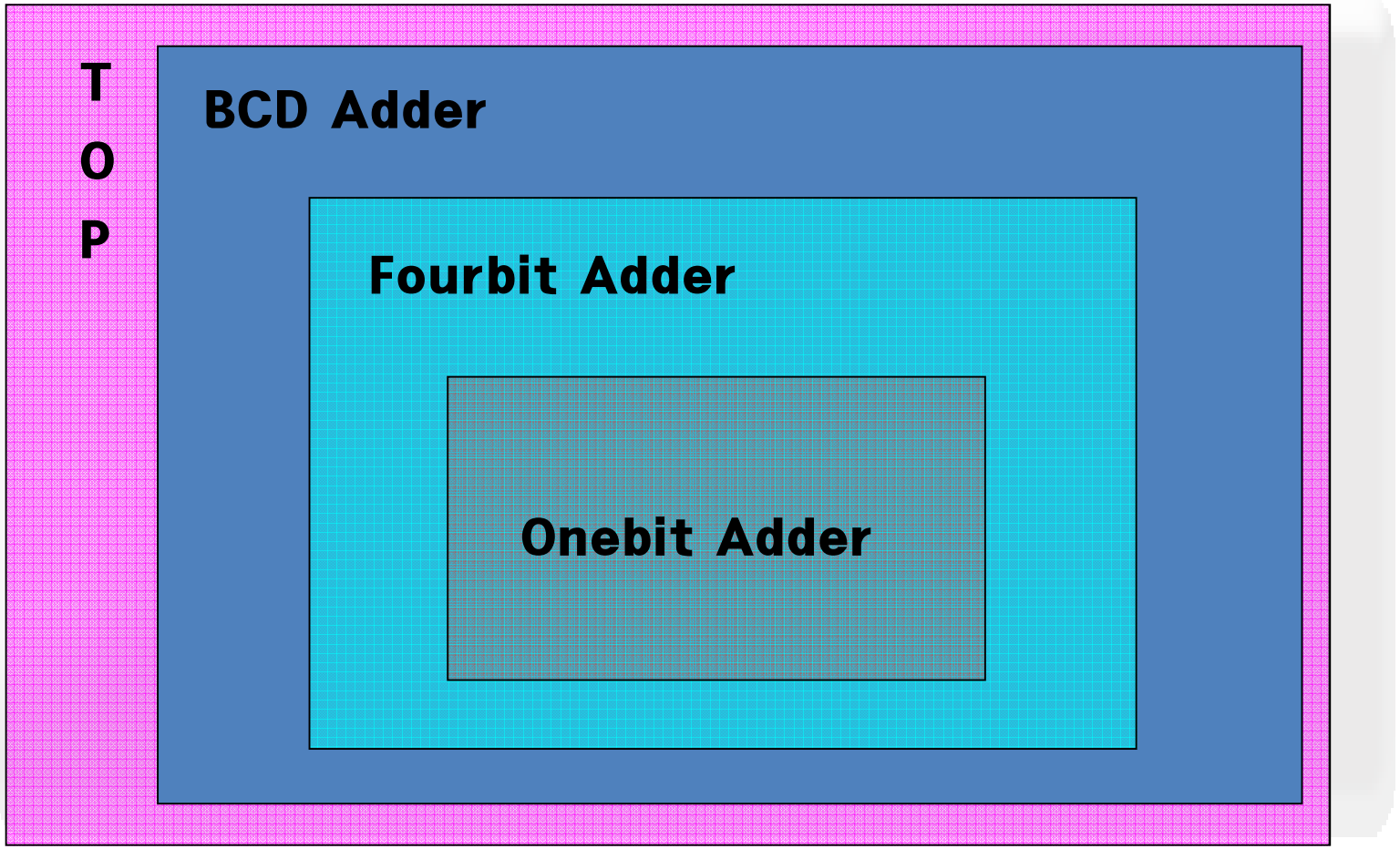
덧셈모듈 구성도

ex) 5 4
+ 3 2
8 6

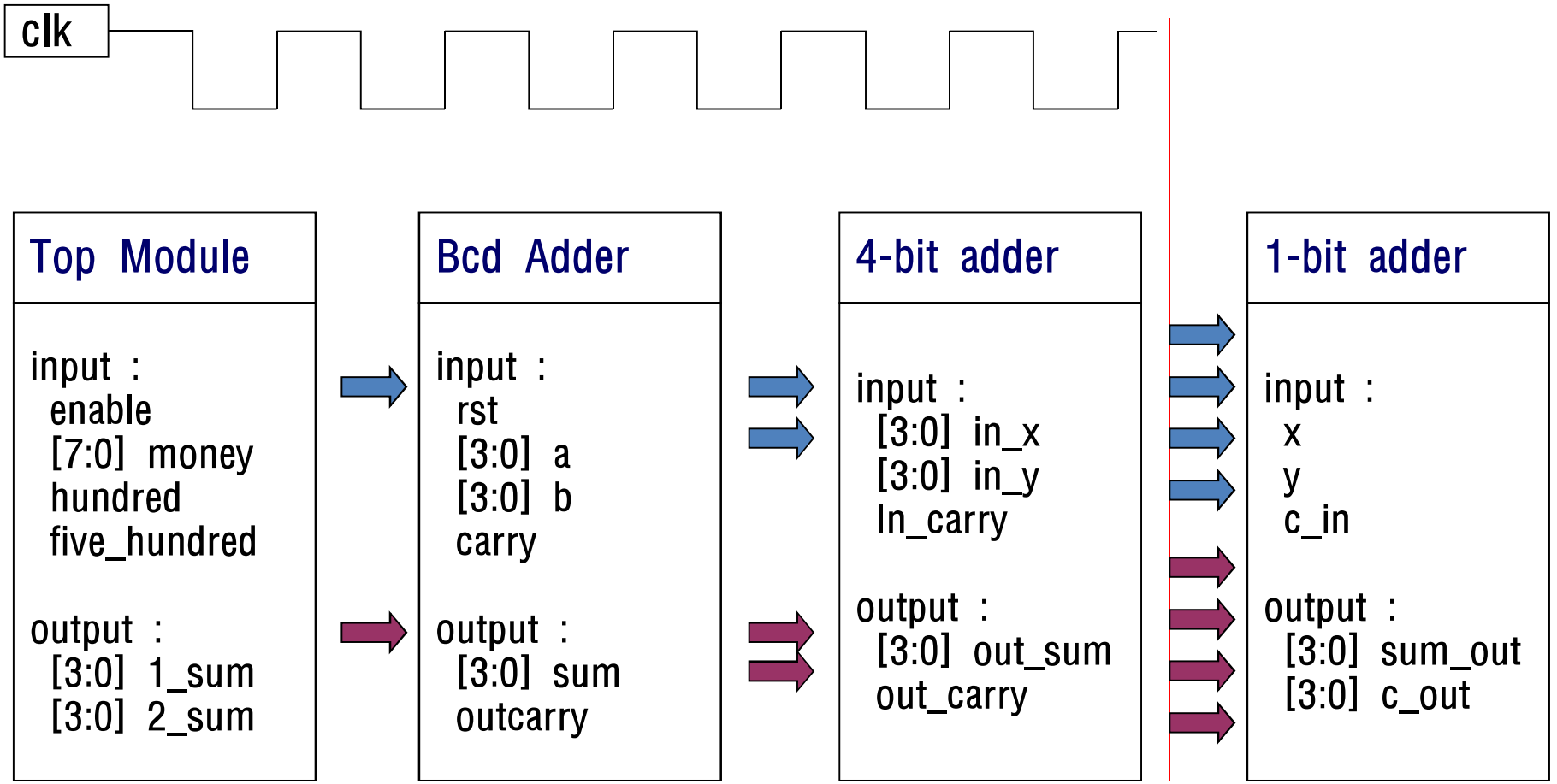
Top Module



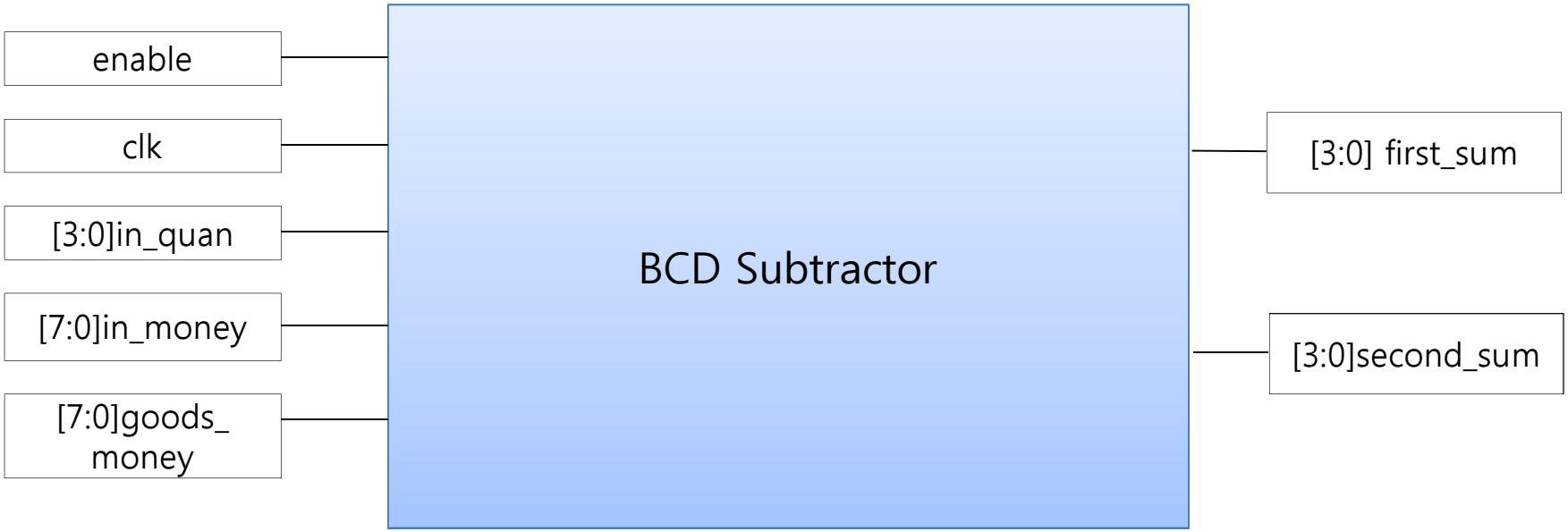
Formation



모듈 관계도


clk
Top Module
input :
enable
[7:0] money
hundred
five_hundred
output :
[3:0] 1_sum
[3:0] 2_sum
Bcd Adder
input :
rst
[3:0] a
[3:0] b
carry
output :
[3:0] sum
outcarry
4-bit adder
input :
[3:0] in_x
[3:0] in_y
In_carry
output :
[3:0] out_sum
out_carry
1-bit adder
input :
x
y
c_in
output :
[3:0] sum_out
[3:0] c_out

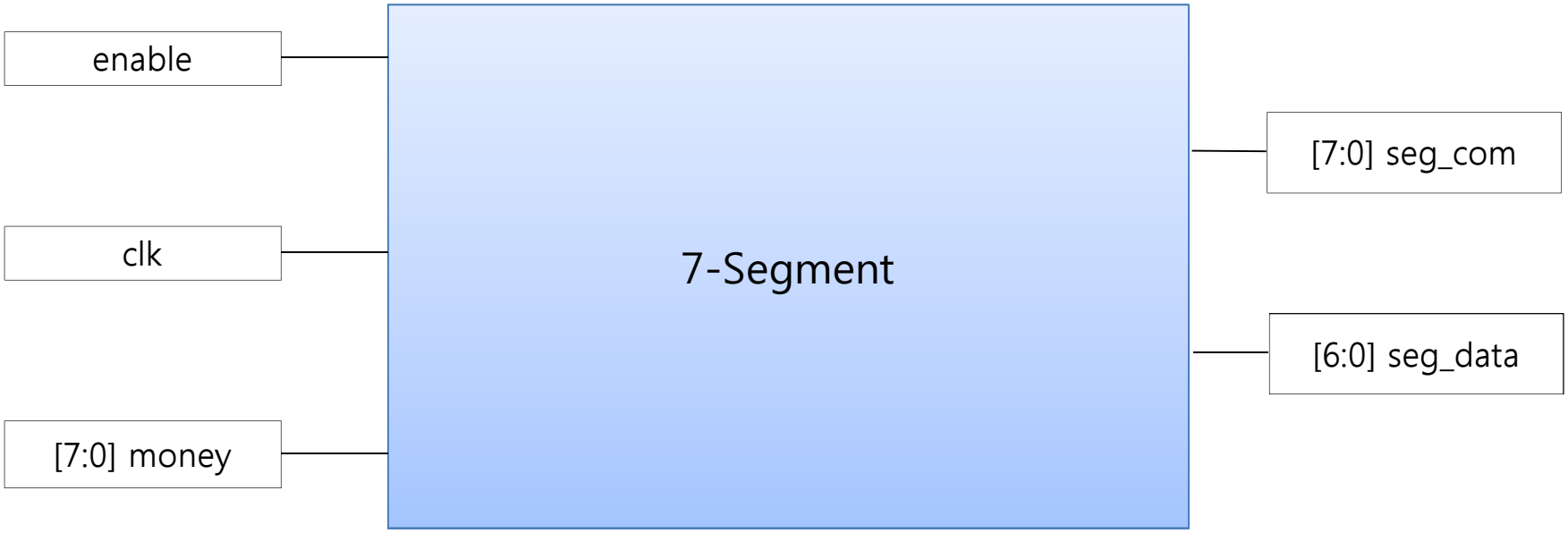
Module (BCD Subtractor)



Module (BCD Subtractor)

Pin Name	I/O	T/S	Description
enable	In	enable (controller)	Enable 신호
clk	In	clk (Top)	CLK
[3:0] in_quan	In	in_quan (controller)	상품의 수량
[7:0]in_money	In	in_money (controller)	컨트롤러의 현재 돈
[7:0]goods_money	In	money (goods)	상품가격
[7:0]out_money	Out	money (controller)	계산 후 잔액
[3:0]out_quan	Out	in_quan (controller)	상품배출 후 수량

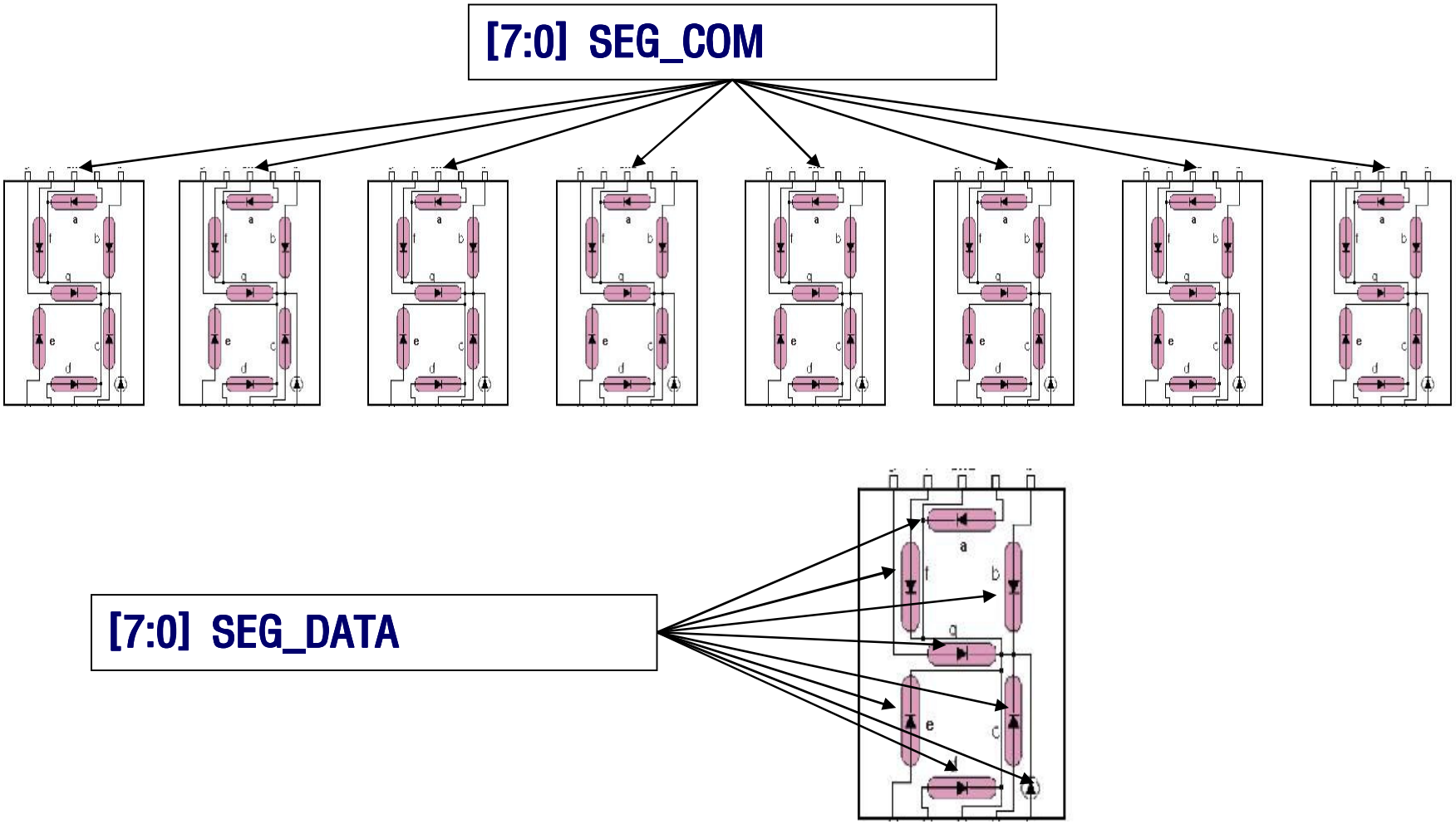
Module (7-Segment)



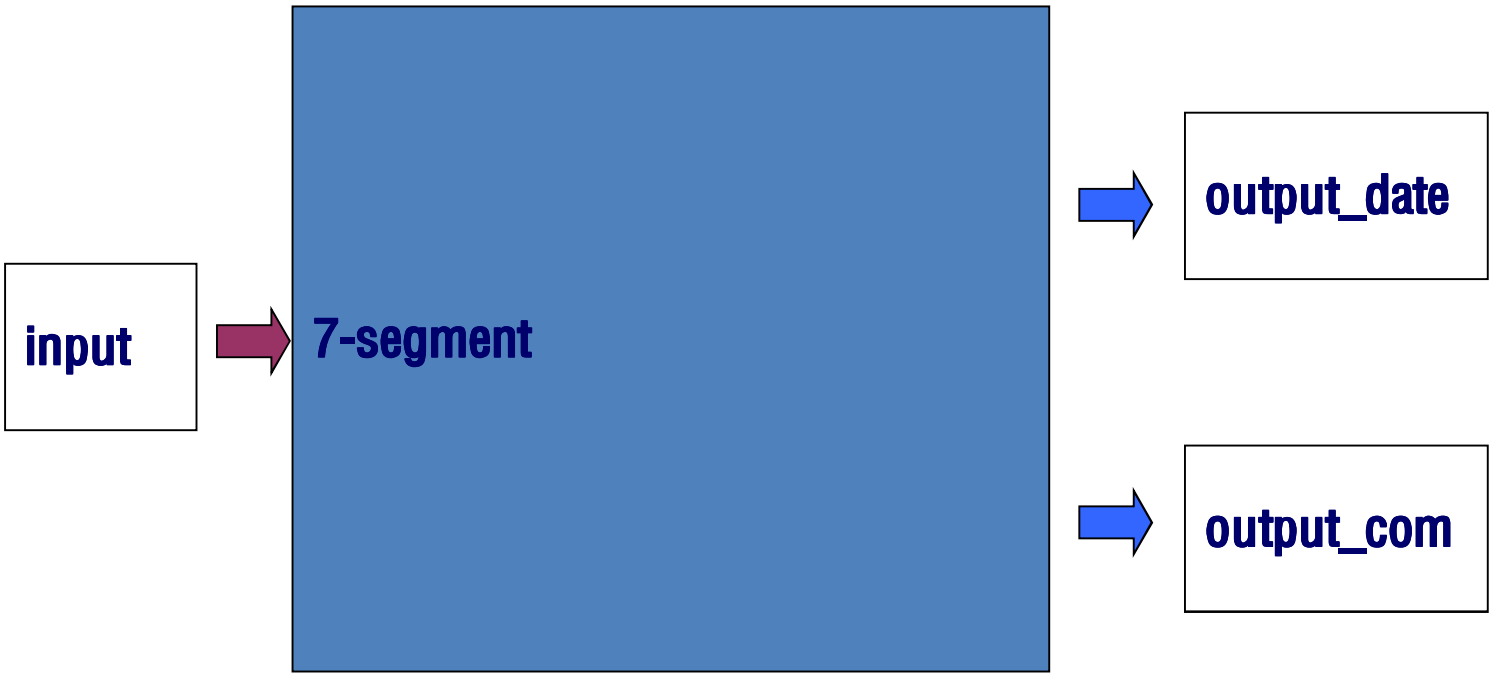
Module (7-Segment)

Pin Name	I/O	T/S	Description
clk	In	clk (Top)	CLK
rst	In	button[5] (Top)	초기화 버튼
[7:0]money	In	money (controller)	출력할 돈의 값
[7:0]seg_com	Out	seg_com (controller)	출력 위치
[6:0]seg_data	Out	seg_data (controller)	출력할 숫자

7-segment 이론



7-segment 구성도



7-segment 알고리즘

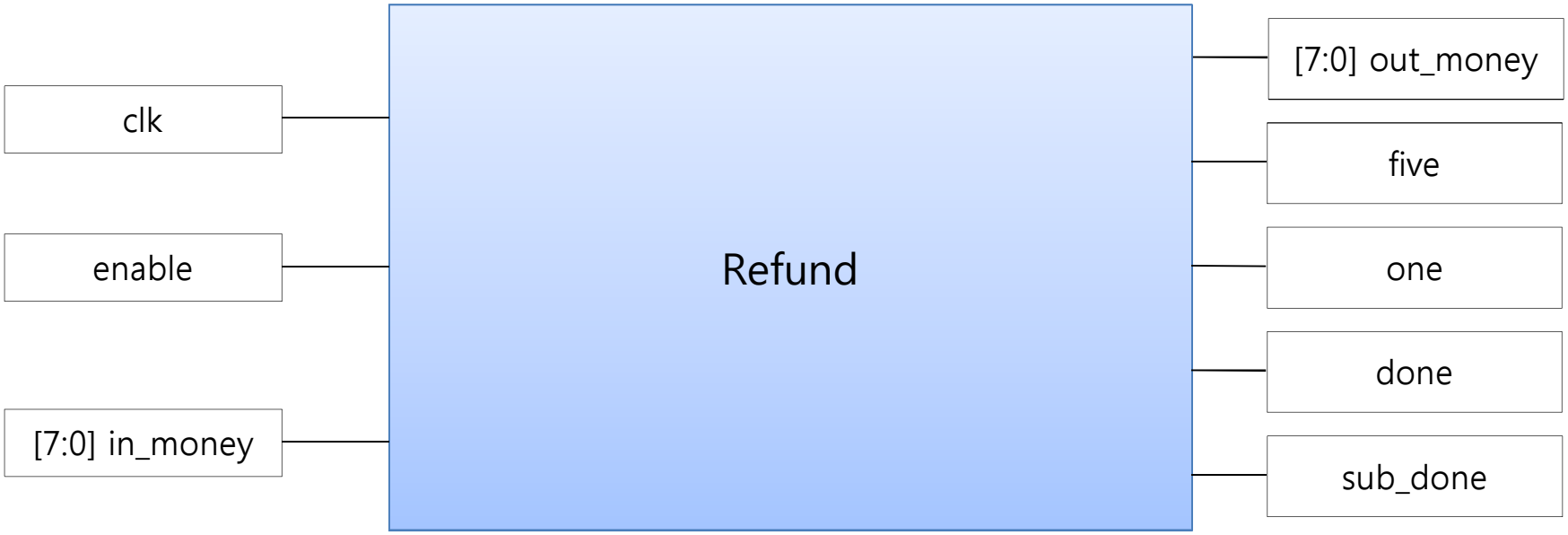
구 분	input	seg_com	seg_out
0	0000	0111_1111	0011_1111
1	0001		0000_0110
2	0010		0101_1011
3	0011		0100_1111
4	0100		0110_0110
5	0101		0110_1101
6	0110		0111_1101
7	0111		0000_0111
8	1000		0111_1111
9	1001		0110_0111

FPGA 보드의 H/W Description

signal	xilinx	description	signal	xilinx	description
SEG_COM[0]	AD2	Segment 1 Select	SEG_DATA[0]	AF5	Segment data A
SEG_COM[1]	W2	Segment 2 Select	SEG_DATA[1]	AE5	Segment data B
SEG_COM[2]	W1	Segment 3 Select	SEG_DATA[2]	AB6	Segment data C
SEG_COM[3]	AB4	Segment 4 Select	SEG_DATA[3]	AA6	Segment data D
SEG_COM[4]	AB3	Segment 5 Select	SEG_DATA[4]	K26	Segment data E
SEG_COM[5]	W6	Segment 6 Select	SEG_DATA[5]	K25	Segment data F
SEG_COM[6]	W5	Segment 7 Select	SEG_DATA[6]	AC2	Segment data G
SEG_COM[7]	W4	Segment 8 Select	SEG_DATA[7]	AC1	Segment data H

보드마다 핀 번호가 다르므로 확인 필수 !!

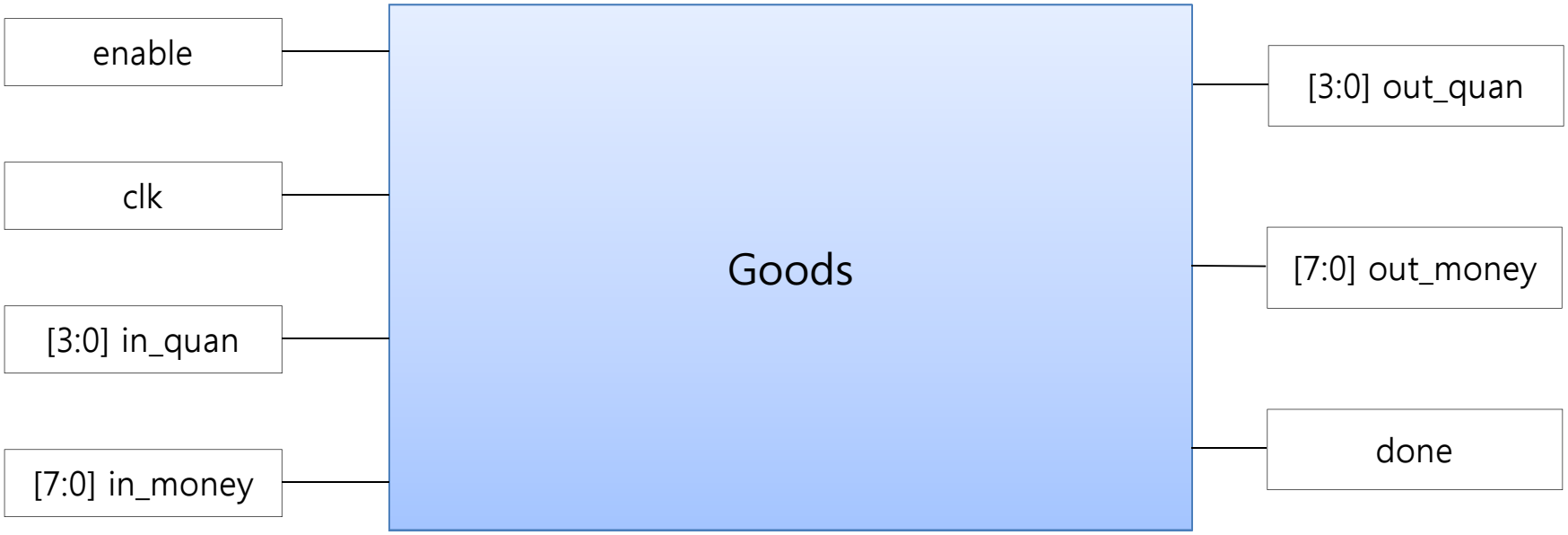
Module (Refund)



Module (Refund)

Pin Name	I/O	T/S	Description
clk	In	clk (Top)	CLK
enable	In	enable (controller)	enable 신호
[7:0]in_money	In	money (controller)	기존의 값
[7:0]out_money	Out	money (controller)	계산 후 값
five	Out	five (controller)	500원 배출 신호
one	Out	one (controller)	100원 배출 신호
Sub_done	Out	finish (controller)	모듈 End 신호
done	Out	finish (controller)	전체 End 신호

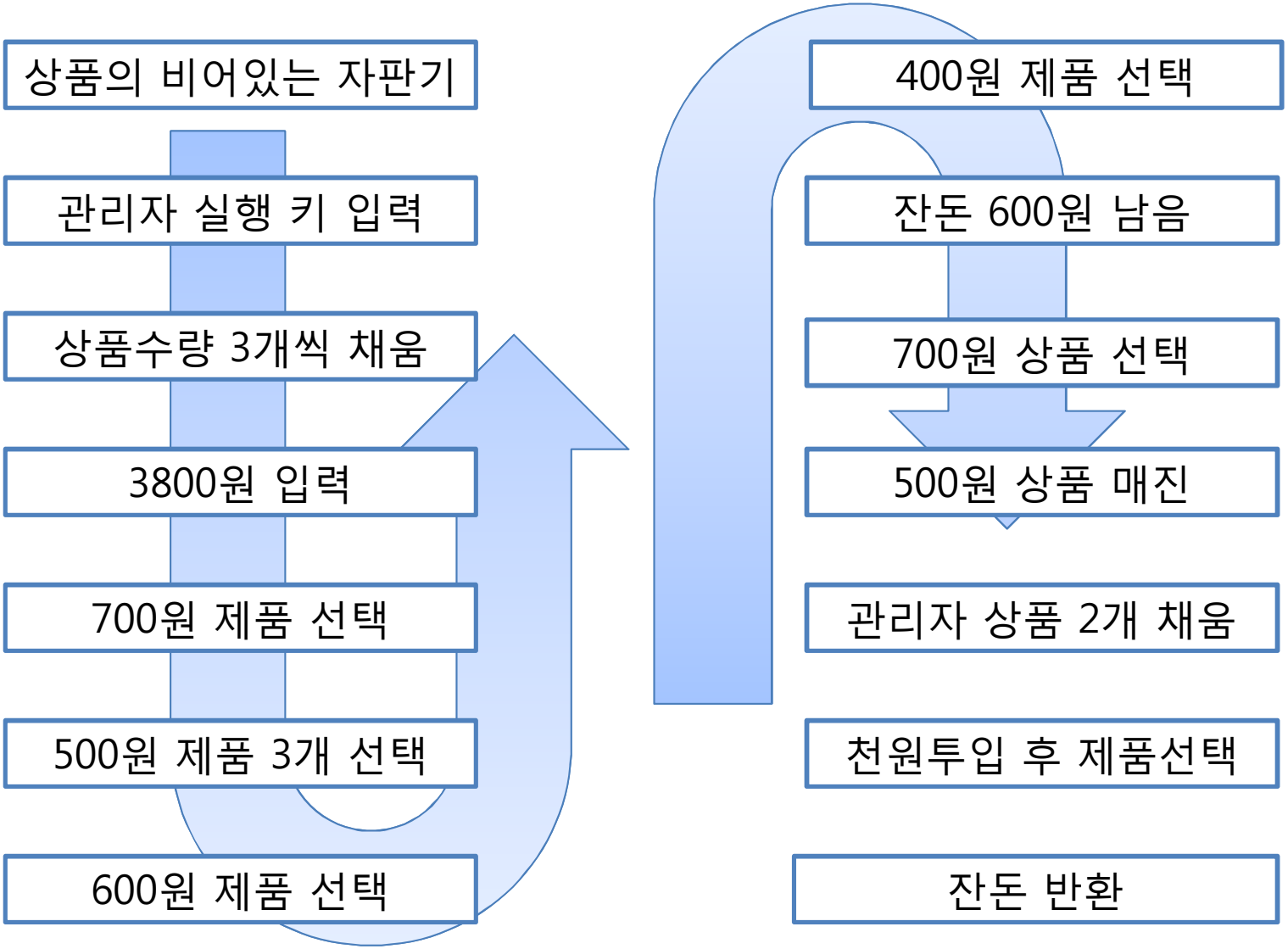
Module (Goods)



Module (Goods)

Pin Name	I/O	T/S	Description
enable	In	enable (controller)	Enable 신호
clk	In	clk (Top)	CLK
[3:0] in_quan	In	quan (controller)	기존 상품 수량
[7:0] in_money	In	money (goods)	기존 상품 가격
[3:0] out_quan	Out	quan (controller)	계산 후 상품 수량
[7:0] out_money	Out	money (controller)	계산 후 상품 가격
done	Out	finish (controller)	End 신호

시뮬레이션 시나리오


상품의 비어있는 자판기
관리자 실행 키 입력
상품수량 3개씩 채움
3800원 입력
700원 제품 선택
500원 제품 3개 선택
600원 제품 선택
400원 제품 선택
잔돈 600원 남음
700원 상품 선택
500원 상품 매진
관리자 상품 2개 채움
천원투입 후 제품선택
잔돈 반환