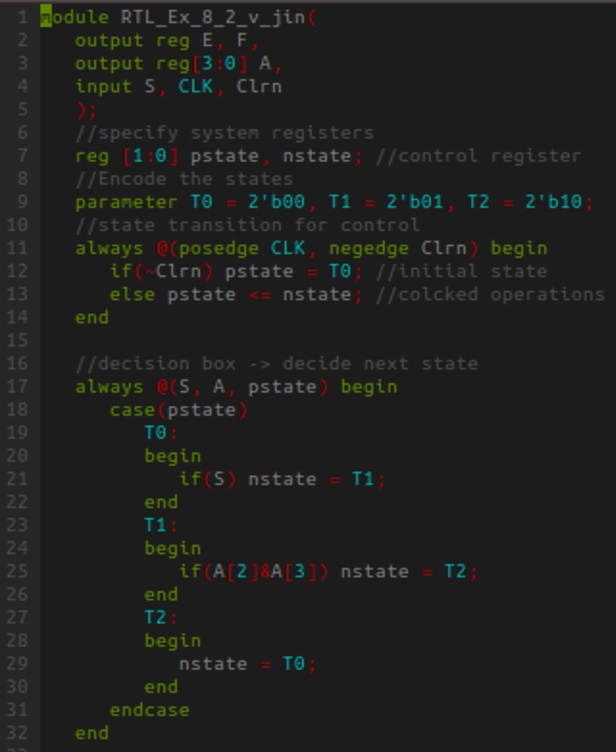
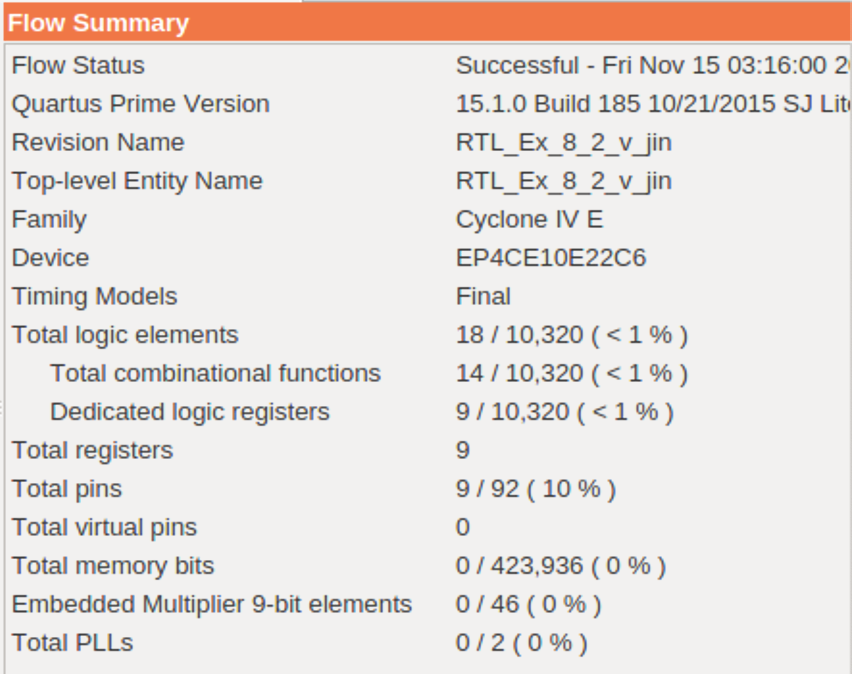
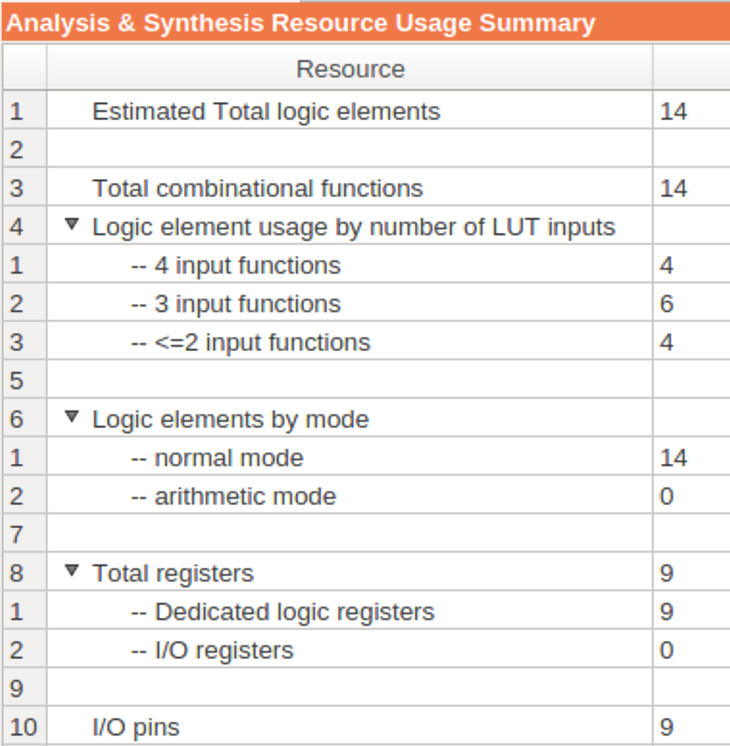
HW#9 디지털 회로 설계 및 언어 월수 9:00~10:15 2015104027 박정진

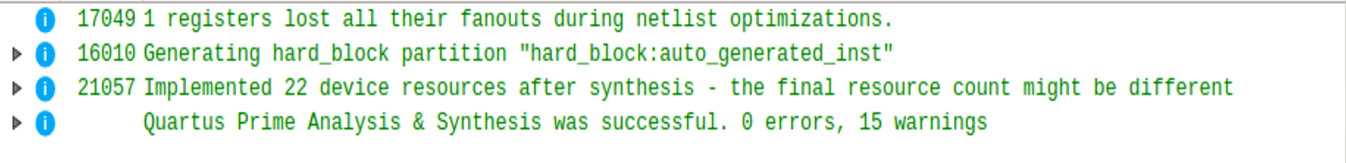
Case 1)

pstate = 0 (blocking assignment)

always (S, A, pstate) no else block





Estimated Total logic elements : 14

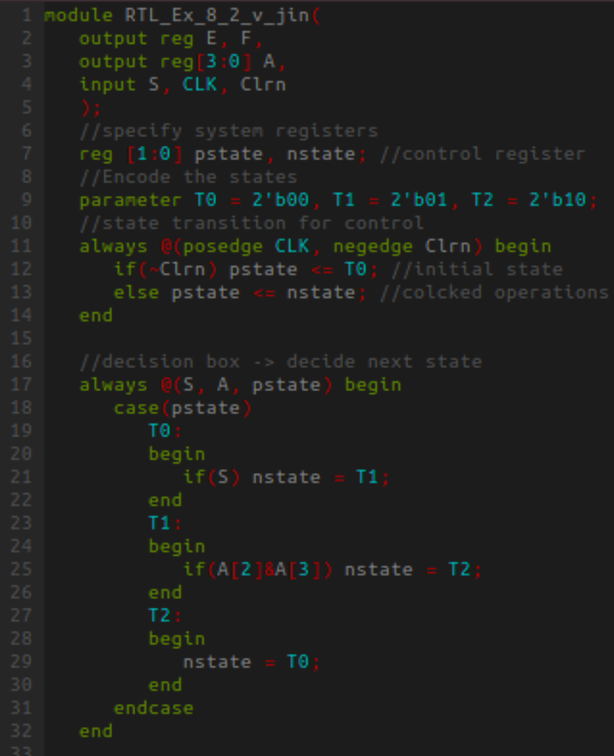
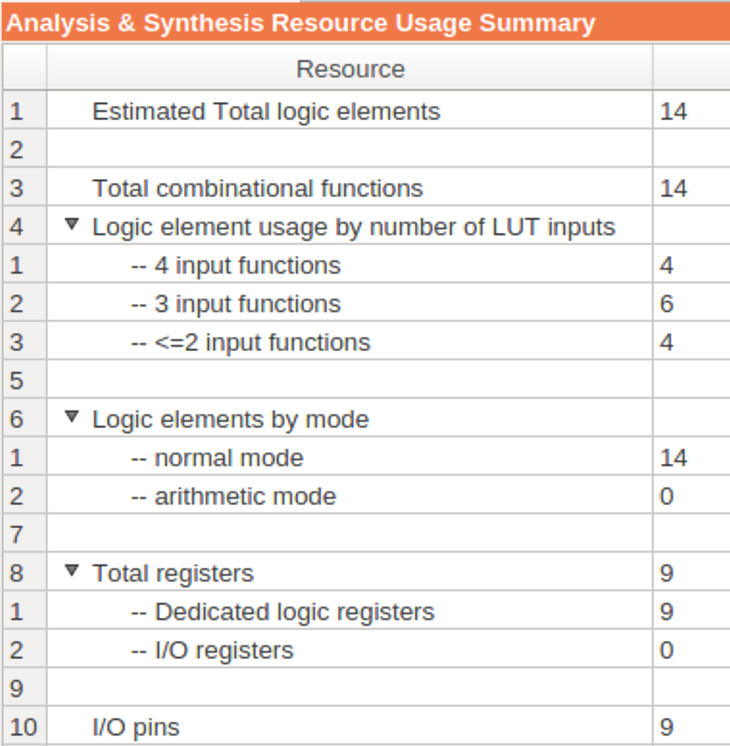
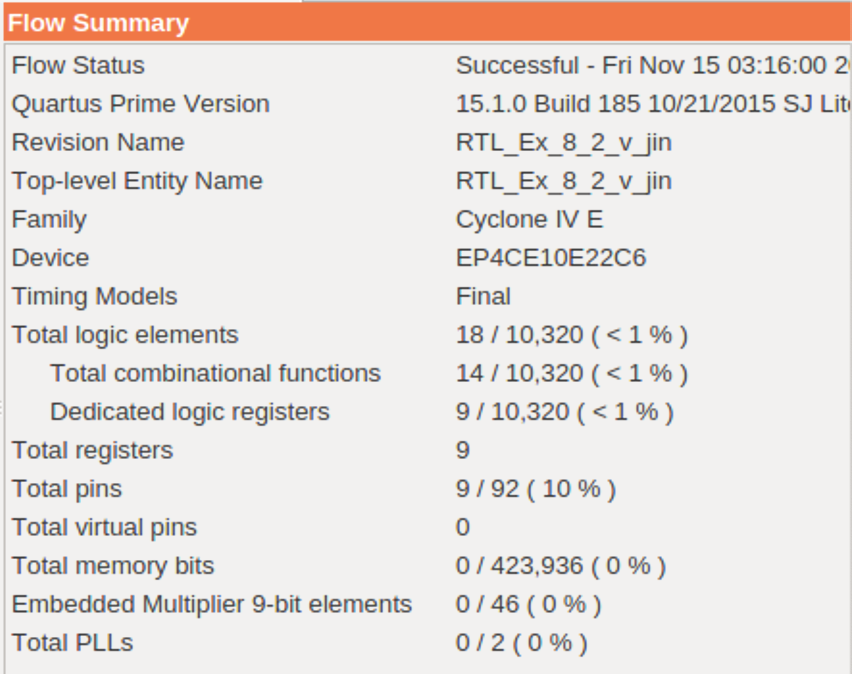
Total Combinational functions : 14

Total Registers : 9

Case 2)

pstate <= 0 (blocking assignment)

always (S, A, pstate) no else block

Estimated Total logic elements : 14

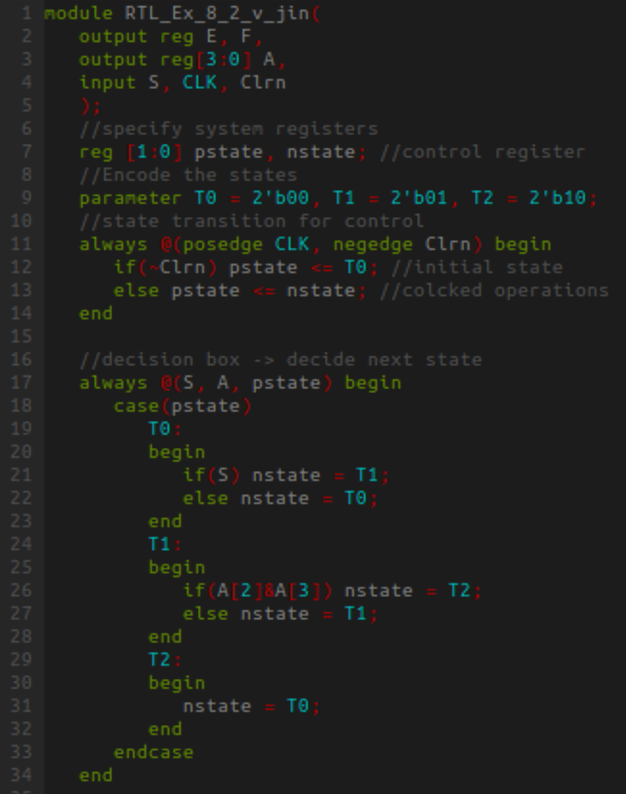
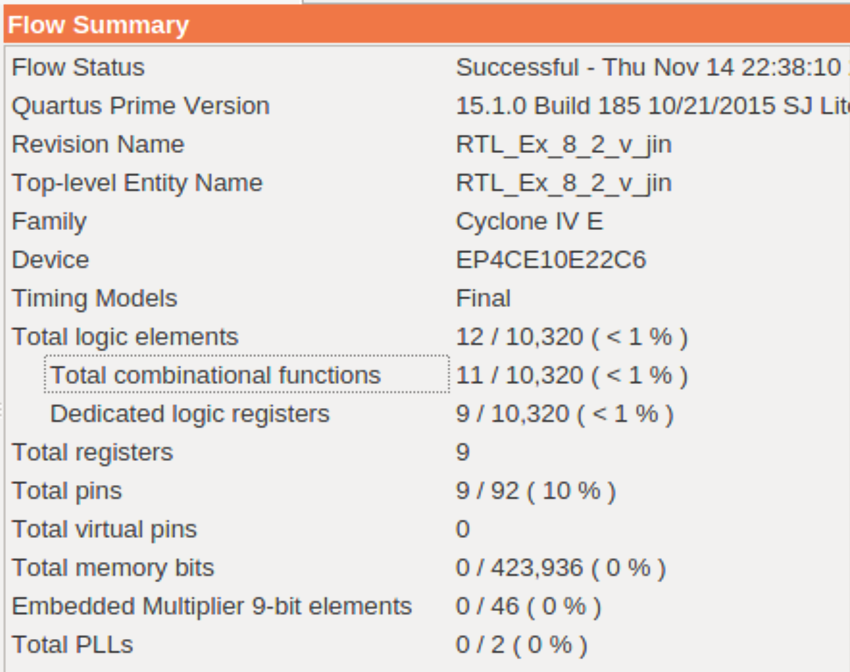
Total Combinational functions : 14

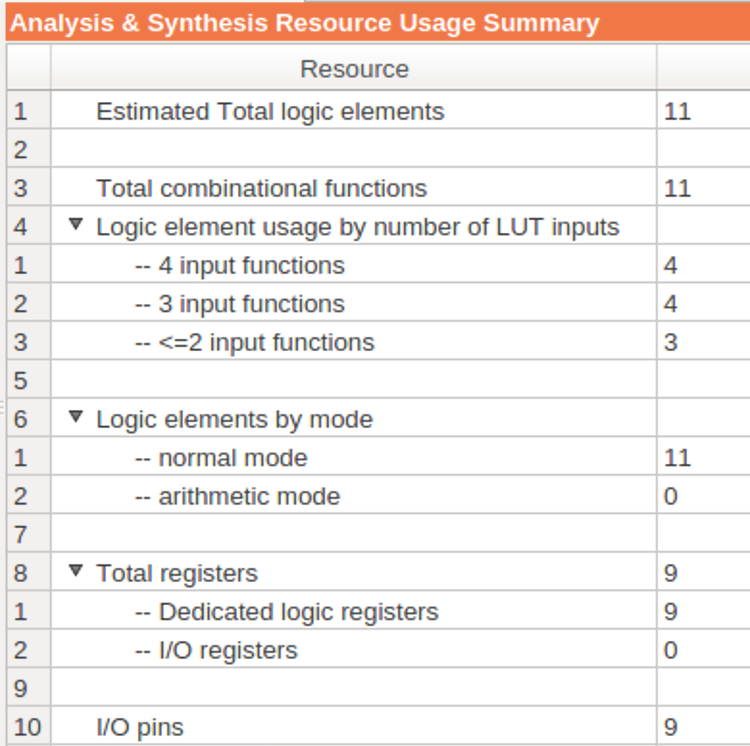
Total Registers : 9

Case 1) 과 차이 없음

Case 3)

always (S, A, pstate) have all else block



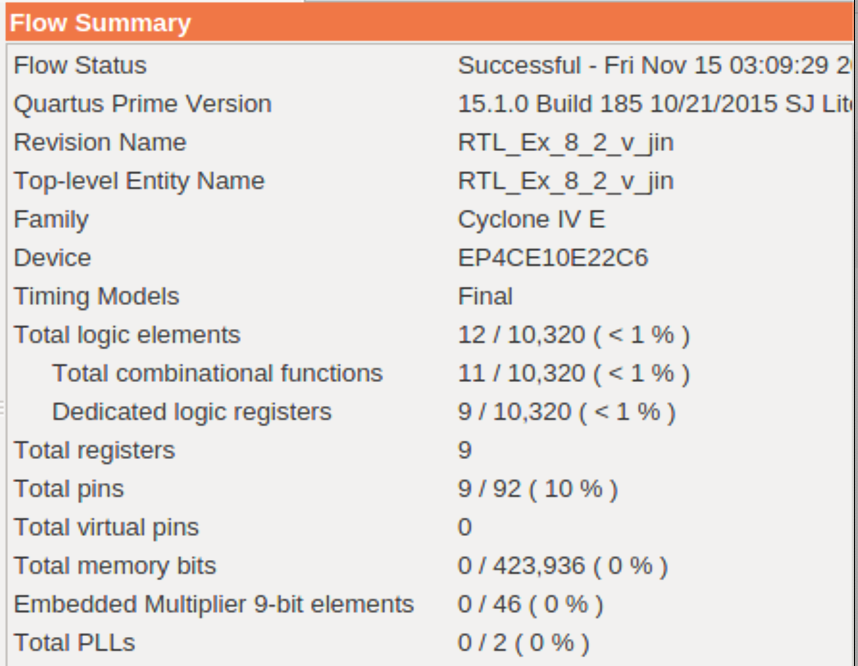
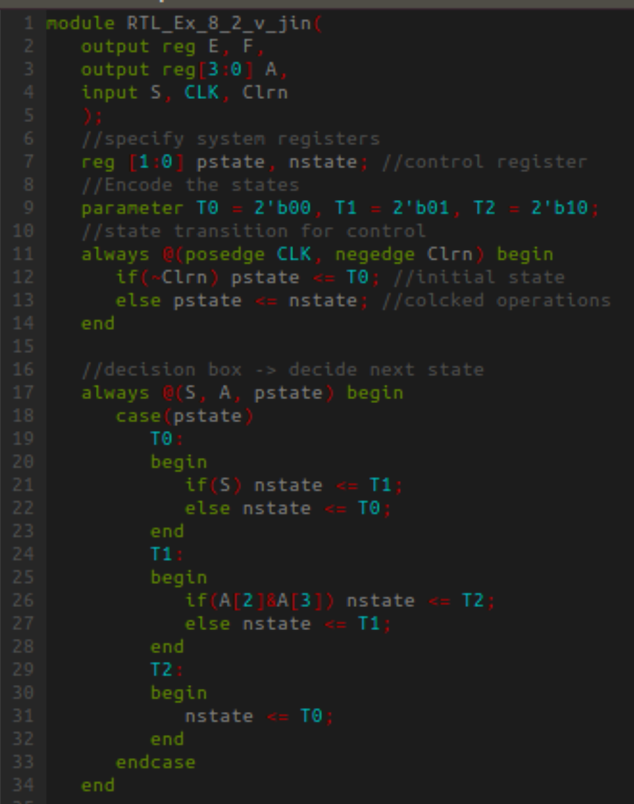
Estimated Total logic elements : 11 (3개 감소)

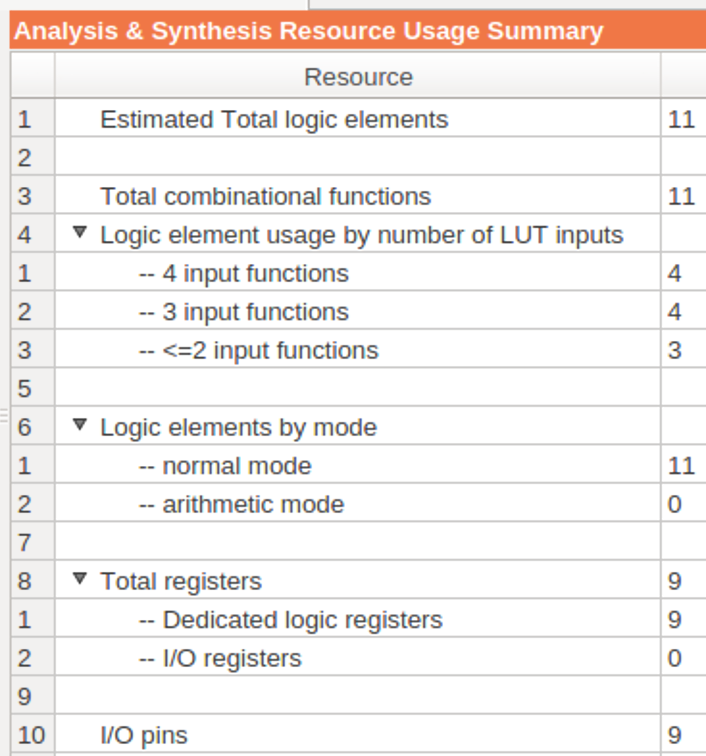
Total Combinational functions : 11

Total Registers : 9

Case 4)

always (S, A, pstate) all non-block assignment





Estimated Total logic elements : 11

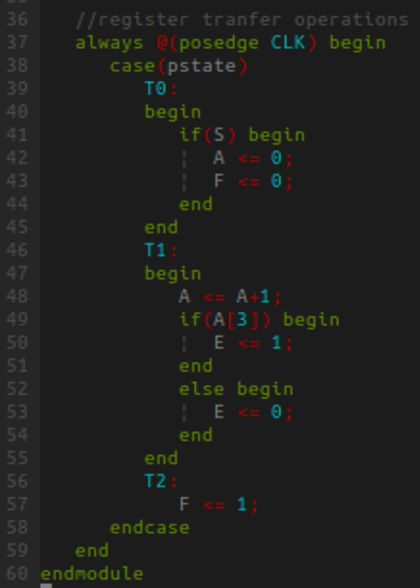
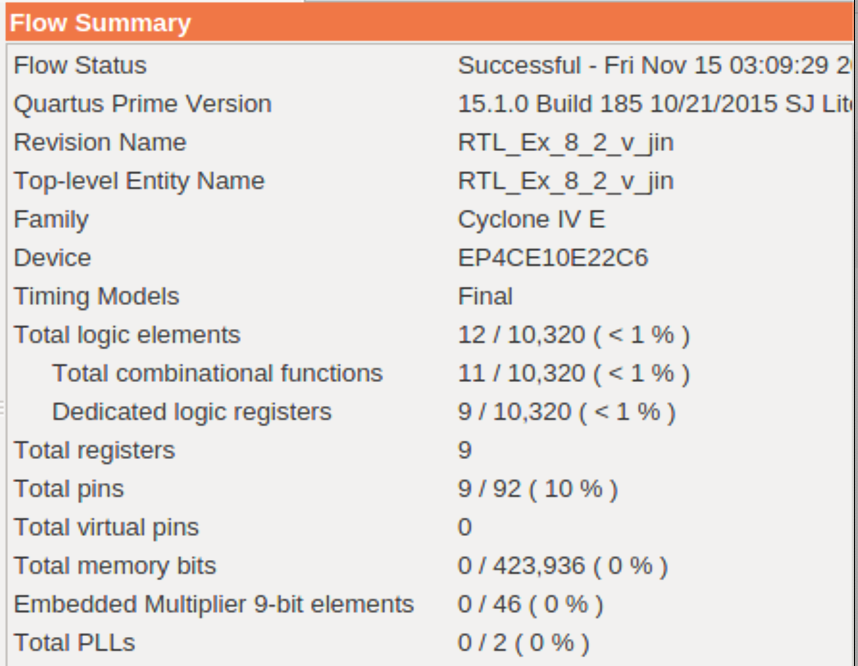
Total Combinational functions : 11

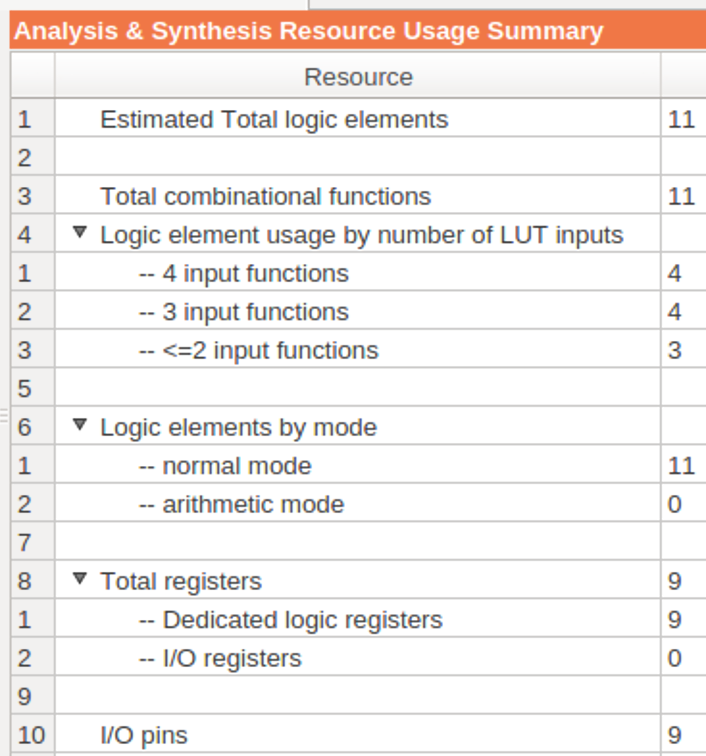
Total Registers : 9

Case 3) 과 차이 없음

Case 5)

always (S, A, pstate) x’b -> decimal integer



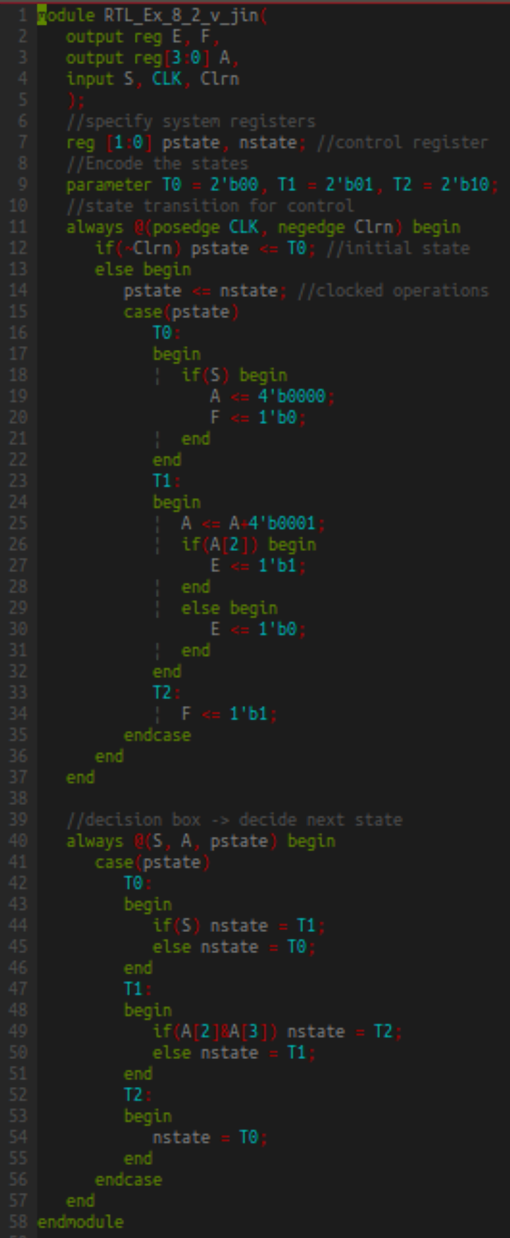
Estimated Total logic elements : 11

Total Combinational functions : 11

Total Registers : 9

Case 3) 과 차이 없음

최종 코드 :

A screenshot of a computer screen

Description automatically generated

A close up of a device

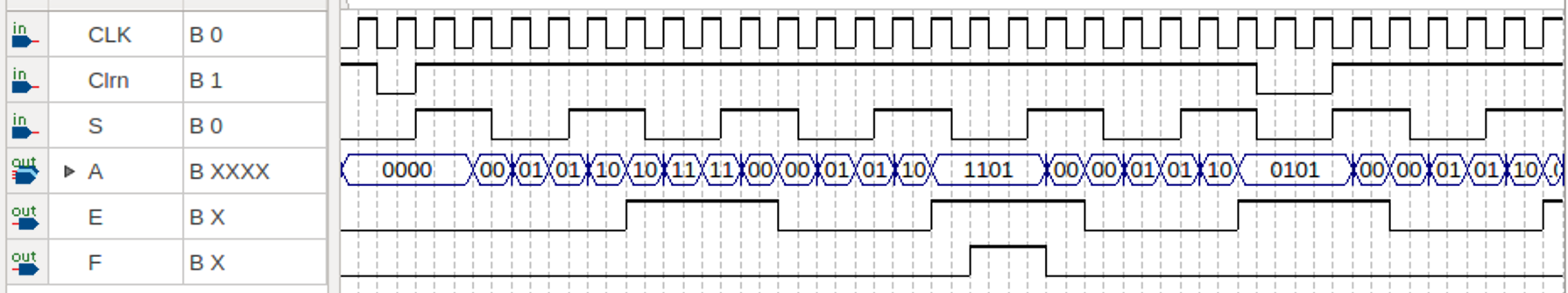
Description automatically generated

Register operation의 동작에서 carry 가 있을 시 A의 값이 변화하면서 A register에 저장된 값이 바뀌며 생기는 delay time이 존재하는 것을 알았고 이와 마찬가지로 Flip-Flop E, F도 해당 값이 바뀌면 flip flop에 의한 delay가 생기는 것을 알 수 있었다.

코드 상에서 로 바꿔도 같았으며 이는 값이flip-flop을 통과하며 생기는 어쩔 수 없는 delay time이라는 것을 알게 되었다. 그래서 이후 combinational logic으로 control unit을 짤 때 combinational 은 기억을 하지 못하므로 나온 결과를 Flip-Flop에 의도적으로 통과 시켜 Register와 같은 Delay time을 주어서 연산을 맞추는 것도 필요 할 것 같다는 생각이 든다.

일반적으로 회로가 돌아가고 첫 edge에서 두 번째 edge는 클리어 신호를 주어 회로의 초기값을 지정한다. 하지만 책에 있는 대로 state operation과 Register operation을 코드 상에서 분리하면 현재 디자인에서는 문제가 없지만 정확하게 이는 시스템 전체를 초기화 시키는 Clrn의 의미가 아니다.(단순히 state만을 초기화) 만약 시스템 전체적으로 초기화를 진행하는 Clrn이라는 의미의 동작을 행하려면 이런 분리된 코드에서는 Register operation에서도 negedge Clrn 을 추가해 초기화를 해야 한다. 그렇지 않으면 Clrn 0일 때에도 Clrn의 신호를 무시한 채 posedge CLK일 때 Register operation은 따로 동작하기에 원하는 시스템 전체적으로 초기화가 진행되지 않을 수 있다. 위의 VWF 결과에서도 확인 할 수 있듯이 Clrn 신호가 들어갔음에도 연산이 수행되는 것을 확인 할 수 있다. 이렇게 되면 resigster operation이 state operation과 다른 상태(초기값이 맞춰지지 않은 상태)에서 동작을 수행하기에 원하는 결과가 나오지 않을 수 있다. (Ex. Register 의 조건이 next state를 결정하는 디자인)

VWF (Timing Sequence 확인)



|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Input | | Main  input | Counter | FF | | Conditions | pstate | nstate | Operations |
| CLK | Clrn | S | A[3:0] | E | F |  |  |  |  |
|  | 1 | 0 | 0000 | 0 | 0 | = 0  = 0 |  |  |  |
|  |  | 0 | 0000 | 0 | 0 | = 0  = 0 |  |  | Asynchronous  reset |
|  | 0 | 0 | 0000 | 0 | 0 | = 0  = 0 |  |  |  |
|  | 1 | 1 | 0000 | 0 | 0 | = 0  = 0 |  |  |  |
|  | 1 | 1 | 0000 | 0 | 0 | = 0  = 0 |  |  |  |
|  | 1 | 1 | 0001 | 0 | 0 | = 0  = 0 |  |  |  |
|  | 1 | 0 | 0010 | 0 | 0 | = 0  = 0 |  |  |  |
|  | 1 | 0 | 0011 | 0 | 0 | = 0  = 0 |  |  |  |
|  | 1 | 1 | 0100 | 0 | 0 | = 1  = 0 |  |  |  |
|  | 1 | 1 | 0101 | 1 | 0 | = 1  = 0 |  |  |  |
|  | 1 | 0 | 0110 | 1 | 0 | = 1  = 0 |  |  |  |
|  | 1 | 0 | 0111 | 1 | 0 | = 1  = 0 |  |  |  |
|  | 1 | 1 | 1000 | 1 | 0 | = 0  = 1 |  |  |  |
|  | 1 | 1 | 1001 | 0 | 0 | = 0  = 1 |  |  |  |
|  | 1 | 0 | 1010 | 0 | 0 | = 0  = 1 |  |  |  |
|  | 1 | 0 | 1011 | 0 | 0 | = 0  = 1 |  |  |  |
|  | 1 | 1 | 1100 | 0 | 0 | = 1  = 1 |  |  |  |
|  | 1 | 1 | 1101 | 1 | 0 | = 1  = 1 |  |  |  |
|  | 1 | 0 | 1101 | 1 | 1 | = 1  = 1 |  |  |  |
|  | 1 | 0 | 1101 | 1 | 1 | = 1  = 1 |  |  |  |
|  | 1 | 1 | 1101 | 1 | 1 | = 1  = 1 |  |  |  |
|  | 1 | 1 | 0000 | 1 | 0 | = 0  = 0 |  |  |  |
|  | 1 | 1 | 0001 | 0 | 0 | = 0  = 0 |  |  |  |
|  | 1 | 0 | 0010 | 0 | 0 | = 0  = 0 |  |  |  |
|  | 1 | 0 | 0011 | 0 | 0 | = 0  = 0 |  |  |  |
|  | 1 | 1 | 0100 | 0 | 0 | = 1  = 0 |  |  |  |
|  | 1 | 1 | 0101 | 1 | 0 | = 1  = 0 |  |  |  |
|  |  | 0 | 0101 | 0 | 0 | = 1  = 0 |  |  | Asynchronous  reset |
|  | 0 | 0 | 0101 | 0 | 0 | = 1  = 0 |  |  |  |
|  | 0 | 0 | 0101 | 0 | 0 | = 1  = 0 |  |  |  |
|  | 1 | 1 | 0101 | 0 | 0 | = 0  = 0 |  |  |  |
|  | 1 | 1 | 0000 | 0 | 0 | = 0  = 0 |  |  |  |
|  | 1 | 1 | 0001 | 0 | 0 | = 0  = 0 |  |  |  |
|  | 1 | 0 | 0010 | 0 | 0 | = 0  = 0 |  |  |  |
| (이후 동작 같음) | | | | | | | | | |