専攻外の方に0802の急所を解説

- 真理値表の理解がポイントです。
- 論理回路 = 組合せ回路 + 順序回路(1.1.5)
  - ここで順序回路は、組合せ回路で作れる。
  - だったら組合せ回路がわかればいい。
  - 組合せ回路は、入力と出力の対応を表にした真理値表で表される。
- 演習 2.2.15 の真理値表が理解できればOK
- 真理値表をHDLで記述しさえすれば、ツールが自動的に LSI (FPGA/ASIC)に回路を作ってくれる。論理圧縮などもすべてツールがやってくれる。(2.2.14-2)
- つまづきやすいのは、順序回路を組合わせ回路に変換する真理値表の作り方(2.3.3)ここさえわかればOK – (FFフリップフロップが状態を保持してる点)

4ビットのバイナリカウンタを考える

0802key.pdf

- バイナリカウンタは、クロックに同期して数を1つずつ増やすカウントをする。
- https://kazuhikoarase.github.io/simcirjs/#-NOucjSMQX5jF_A6zc-M
- クロックに同期するという意味は、クロック入力が 0->1 に変化する(立ち上がり)で値が変化するという意味。
- よって真理値表の出力は0000->1111になる。
- 最後1111に達すると0000に戻る。(1000に進むの下の4ビットは0000という表示になる。)
- クロックに同期して動作するので同期式順序回路である。

入力		出力
	CLOCK	0000
	CLOCK	0001
	CLOCK	0010
	CLOCK	0011
	CLOCK	0100
	CLOCK	0101
	CLOCK	0110
	CLOCK	0111
	CLOCK	1000
	CLOCK	1001
	CLOCK	1010
	CLOCK	1011
	CLOCK	1100
	CLOCK	1101
	CLOCK	1110
	CLOCK	1111
	CLOCK	0000

順序回路の定義は？

- 順序回路は、現在の内部状態と次の入力によって次の出力が定まる回路である。
- 現在の内部状態は、フリップフロップに値が保持されている。そして次の入力は、クロックである。
- 真理値表の出力欄は、”次の状態”を決定するための入力を求めている。2行目の値が0001になるためには、前のカウンタの値の内部状態が入力されないといけない。この値は、0000である。

入力		次の出力
	次のCLOCK	0000
0000	次のCLOCK	0001
	次のCLOCK	0010
	次のCLOCK	0011
	次のCLOCK	0100
	次のCLOCK	0101
	次のCLOCK	0110
	次のCLOCK	0111
	次のCLOCK	1000
	次のCLOCK	1001
	次のCLOCK	1010
	次のCLOCK	1011
	次のCLOCK	1100
	次のCLOCK	1101
	次のCLOCK	1110
	次のCLOCK	1111
	次のCLOCK	0000

順序回路の真理値表は？

- よって順序回路を表現する真理値表の入力は、現在の状態と次のクロックになる。
- この入力によって次の状態=出力がきまる。
- そうすると次の状態が0000になるための現在の状態はカウンタの動作から1111となる。
- このように次の出力は入力の状態がひとつ前にずれた”現在の状態”になる。
- このズレ入力がわかればOK

0802key.pdf

入力		出力
現在の状態	次のCLOCK	次の状態
1111	次のCLOCK	0000
0000	次のCLOCK	0001
	次のCLOCK	0010
	次のCLOCK	0011
	次のCLOCK	0100
	次のCLOCK	0101
	次のCLOCK	0110
	次のCLOCK	0111
	次のCLOCK	1000
	次のCLOCK	1001
	次のCLOCK	1010
	次のCLOCK	1011
	次のCLOCK	1100
	次のCLOCK	1101
	次のCLOCK	1110
	次のCLOCK	1111
	次のCLOCK	0000

真理値表の完成

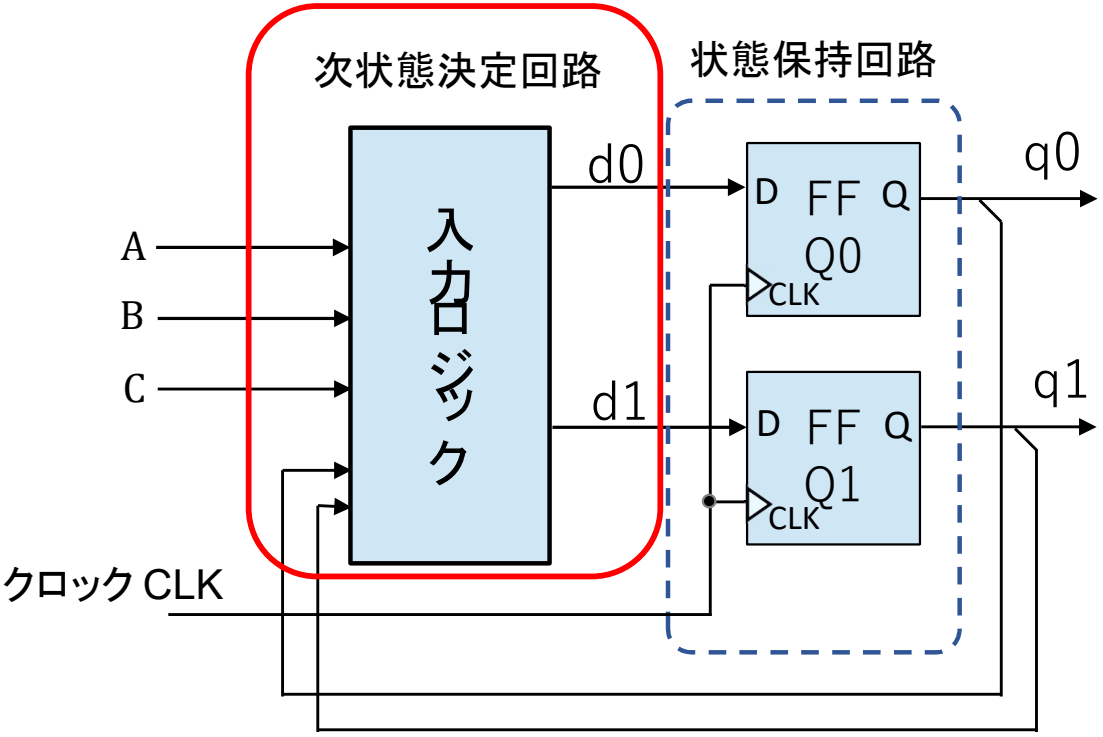
- よって順序回路を表現する真理値表の入力は、現在の状態と次のクロックになる。
- この入力によって次の状態=出力がきまる。
- そうすると次の状態が0000になるための現在の状態はカウンタの動作から1111となる。
- このように次の出力は入力の状態がひとつ前にずれた”現在の状態”になる。
- このズレ入力がわかればOK
- 最後の行は最初の行と同じなので削除してある。
- 2.3.3 のスライドをもう一度見てみよう、

0802key.pdf

入力		出力
現在の状態	次のCLOCK	次の状態
1111	次のCLOCK	0000
0000	次のCLOCK	0001
0001	次のCLOCK	0010
0010	次のCLOCK	0011
0011	次のCLOCK	0100
0100	次のCLOCK	0101
0101	次のCLOCK	0110
0110	次のCLOCK	0111
0111	次のCLOCK	1000
1000	次のCLOCK	1001
1001	次のCLOCK	1010
1010	次のCLOCK	1011
1011	次のCLOCK	1100
1100	次のCLOCK	1101
1101	次のCLOCK	1110
1110	次のCLOCK	1111

2.3.3 順序回路の構成と真理値表

- 同期式順序回路は状態保持FFとその前段のロジックから構成される。前段は、”次の状態”を決定する組合せ回路である。



- 赤枠の部分が組合せ回路であるからこの部分の動作を真理値表で表記すればよい。
- 出力は、次のクロックの到来でFFに格納されるので入力と出力はクロックを境に現在と次の状態に区分されることになる。

入力					出力	
			現在の状態		次の状態	
A	B	C	q0	q1	d0(次のq0)	d1(次のq1)

ステートマシン

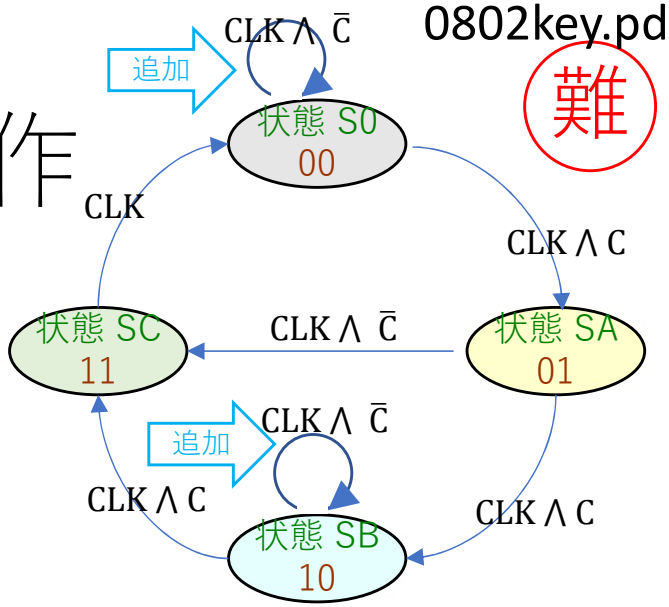
- クロックに同期して状態を変化させるものがステートマシンである。その意味でバイナリカウンタもステートマシンのひとつである。
- あらゆるステートマシン、順序回路は状態遷移図で書ける。そしてこれはそのまま真理値表になる。
- 状態遷移図と真理値表の対応がわかればOK
- 2.4.2 のスライドをもう一度見てみよう。このスライドでは、入力のクロックが省略されている。C が制御入力である。次の出力は、現在の状態と制御入力できまる。この関係はただちに真理値表で表される。

斜体文字部は0802講義時には理解しなくてよい。

0802key.pdf

2.4.2 ステートマシンの3動作

難



- 同期式ステートマシンの動作は、GO STOP JUMPの3つである。この動作は、入力条件によって制御される。GOは次の状態への遷移、STOPは現在の状態にとどまり、JUMPは指定した状態に遷移する。
- 右図は2ビットのバイナリ・カウンタで、C入力を持つ。Cが1のときはクロックに同期して+1カウントアップする。状態SA(01)ときCが0のときは、状態SC(11)にジャンプする。
- このように同期式ステートマシンでは、FFの駆動回路は必ず真理値表で表現できる。よって論理式がただちに求まる。
- *状態SA(01)からSC(11)の遷移路は、ハミング距離が1であることを注意しておこう。このときCはこの系のクロックに同期しない入力(非同期入力)が可能である。(この理由はイントリンシックハザードが発生しないためである。)*
- *後の項で取り上げる非同期入力とは異なるので注意しよう。*

追加

	入力 (CLKは省略) 現在の状態			出力 次の状態		
	d0	d1	C	d0	d1	
S0	0	0	0	0	***0***	修正
	0	0	1	0	1	
SA	0	1	0	1	1	
	0	1	1	1	0	
SB	1	0	0	1	***0***	修正
	1	0	1	1	1	
SC	1	1	0	0	0	
	1	1	1	0	0	