Binary-to-BCD Converter

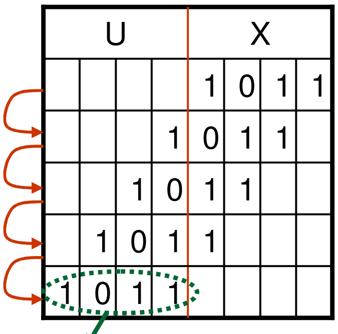
Double-Dabble Binary-to-BCD
Conversion Algorithm

Basic Idea

- Y←X, X is a 4-bit binary number
  - Y is a 4-bit binary number (Binary to binary)
    ⇒ can be done by only shifting
    ex: 1011 ← 1011 (shift left 4 times)
  - Y is a BCD number (Binary to BCD)
    ∵X: 0000~1111,
    ∴Y: 00~15 (two BCD digits, at least 5 bits))
    ex: 01000 ← 1000
    ? ← 1011

Y				X			
				1	0	1	1
			1	0	1	1	
		1	0	1	1		
	1	0	1	1			
1	0	1	1				

```
if (U > 4)
   then U=U+3;
Shift left;
```



U				X			
				1	0	1	1
			1	0	1	1	
		1	0	1	1		
	1	0	1	1			
1	0	1	1				

Shift left

U ← U*2+X[3]

Out of range
U=U+6;

if (U > 4) then U will be
Out of range after “shift left”

Double-Dabble Binary-to-BCD Conversion Algorithm

Shift and Add-3 Algorithm (consider 8-bit binary)

1. Shift the binary number left one bit.
2. If 8 shifts have taken place, the BCD number is in the *Hundreds*, *Tens*, and *Units* column.
3. If the binary value in any of the BCD columns is 5 or greater, add 3 to that value in that BCD column.
4. Go to 1.

Example:

8 bits

Operation	Hundreds	Tens	Units	Binary	
HEX				F	F
Start				1 1 1 1	1 1 1 1

Steps to convert an 8-bit binary number to BCD

Operation	Hundreds	Tens	Units	Binary	
HEX				F	F
Start				1 1 1 1	1 1 1 1
Shift 1			1	1 1 1 1	1 1 1
Shift 2			1 1	1 1 1 1	1 1
Shift 3			1 1 1	1 1 1 1	1
Add 3			1 0 1 0	1 1 1 1	1
Shift 4		1	0 1 0 1	1 1 1 1	
Add 3		1	1 0 0 0	1 1 1 1	
Shift 5		1 1	0 0 0 1	1 1 1	
Shift 6		1 1 0	0 0 1 1	1 1	
Add 3		1 0 0 1	0 0 1 1	1 1	
Shift 7	1	0 0 1 0	0 1 1 1	1	
Add 3	1	0 0 1 0	1 0 1 0	1	
Shift 8	1 0	0 1 0 1	0 1 0 1		
BCD	2	5	5		

Example of converting hex E to BCD

Operation	Tens	Units	Binary
HEX			E
Start			1 1 1 0
Shift 1		1	1 1 0
Shift 2		1 1	1 0
Shift 3		1 1 1	0
Shift 4		1 1 1 0	
6		0 1 1 0	
Add 6	1	0 1 0 0	
BCD	1	4	

Steps to convert a 6-bit binary number to BCD

1. Clear all bits of *z* to zero
2. Shift *B* left 3 bits

   $z[8:3] = B[5:0];$
3. Do 3 times

   if *Units* > 4

   then add 3 to *Units*

   (note: *Units* = $z[9:6]$)

   Shift *z* left 1 bit
4. *Tens* = $P[6:4] = z[12:10]$

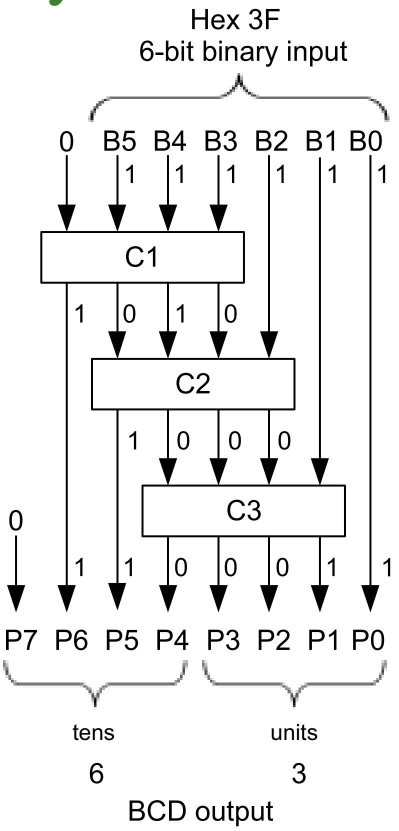
   Units = $P[3:0] = z[9:6]$

Operation	Tens	Units	Binary
B			5 4 3 2 1 0
HEX			3 F
Start			1 1 1 1 1 1
Shift 1		1	1 1 1 1 1
Shift 2		1 1	1 1 1 1
Shift 3		1 1 1	1 1 1
Add 3		1 0 1 0	1 1 1
Shift 4	1	0 1 0 1	1 1
Add 3	1	1 0 0 0	1 1
Shift 5	1 1	0 0 0 1	1
Shift 6	1 1 0	0 0 1 1	
BCD	6	3	
P	7 4	3 0	
z	13 10	9 6	5 0

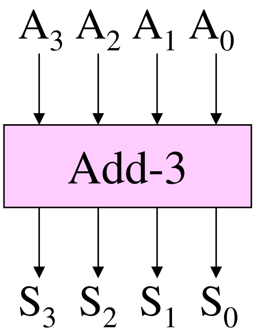
How to implement?

Steps to convert a 6-bit binary number to BCD (Cont'd)

Operation	Tens	Units	Binary
B			5 4 3 2 1 0
HEX			3 F
Start			1 1 1 1 1 1
Shift 1		1	1 1 1 1 1
Shift 2		1 1	1 1 1 1
Shift 3		1 1 1	1 1 1
Add 3		1 0 1 0	1 1 1
Shift 4	1	0 1 0 1	1 1
Add 3	1	1 0 0 0	1 1
Shift 5	1 1	0 0 0 1	1
Shift 6	1 1 0	0 0 1 1	
BCD	6	3	
P	7 4	3 0	
z	13 10	9 6	5 0



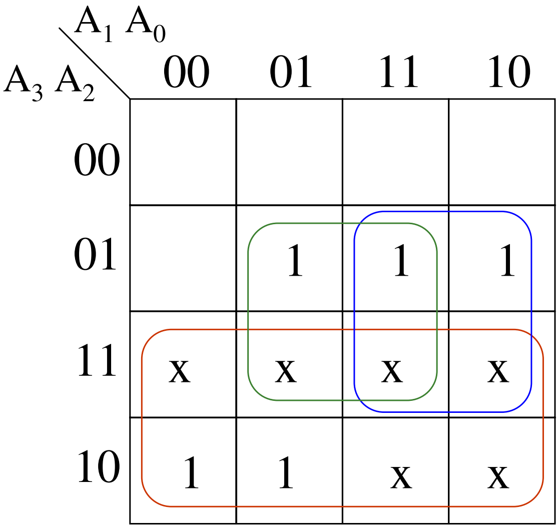
Truth table for Add-3 Module



A_3	A_2	A_1	A_0	S_3	S_2	S_1	S_0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	0
0	0	1	1	0	0	1	1
0	1	0	0	0	1	0	0
0	1	0	1	1	0	0	0
0	1	1	0	1	0	0	1
0	1	1	1	1	0	1	0
1	0	0	0	1	0	1	1
1	0	0	1	1	1	0	0
1	0	1	0	x	x	x	x
1	0	1	1	x	x	x	x
1	1	0	0	x	x	x	x
1	1	0	1	x	x	x	x
1	1	1	0	x	x	x	x
1	1	1	1	x	x	x	x

K-Map for S_3

A_3	A_2	A_1	A_0	S_3	S_2	S_1	S_0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	0
0	0	1	1	0	0	1	1
0	1	0	0	0	1	0	0
0	1	0	1	1	0	0	0
0	1	1	0	1	0	0	1
0	1	1	1	1	0	1	0
1	0	0	0	1	0	1	1
1	0	0	1	1	1	0	0
1	0	1	0	x	x	x	x
1	0	1	1	x	x	x	x
1	1	0	0	x	x	x	x
1	1	0	1	x	x	x	x
1	1	1	0	x	x	x	x
1	1	1	1	x	x	x	x



A_3A_2 \ A_1A_0	00	01	11	10
00				
01		1	1	1
11	x	x	x	x
10	1	1	x	x

$S_3 = A_3+A_2A_0+A_2A_1$

$S_2 = A_3A_0+A_2A_1'A_0'$

$S_1 = A_3A_0'+A_2'A_1+A_1A_0$

$S_0 = A_3A_0'+A_3'A_2'A_0+A_2A_1A_0'$

exercise

- Design a Verilog module to convert an 8-bit binary number to the BCD form.

```
module Binary_to_BCD_8(P,B);
output [9:0] P; //BCD form of B
input [7:0] B;
. . .
endmodule
```