Assignment 2: Combinational Double-Dabble

Group 20

January 24, 2021

1 4-bit binary to BCD convertor

Let the number be

$$b_0 b_1 b_2 b_3$$

and the BCD value be

$$d_1 d_2 d_3 d_4 d_5$$

In the 4 bit binary to BCD converter, we used one unit of 3-adders

1. The output cannot exceed 5 until the three bits ($b_1 b_2 b_3$) are shifted in
2. Hence, we only need one 3 adder for when $b_1 b_2 b_3$ are shifted in
3. After which only b_0 is left, which is the last bit and remains unchanged

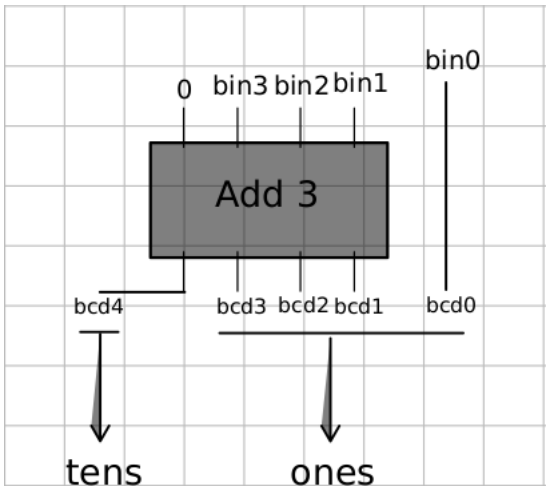

Figure 1: 7 Bit binary to BCD convertor

2 6-bit binary to BCD convertor

Let the number be

$$b_0 b_1 b_2 b_3 b_4 b_5$$

and the BCD value be

$$d_1 d_2 d_3 d_4 d_5 d_6 d_7 d_8$$

In the 6 bit binary to BCD converter, we used three units of 3-adders

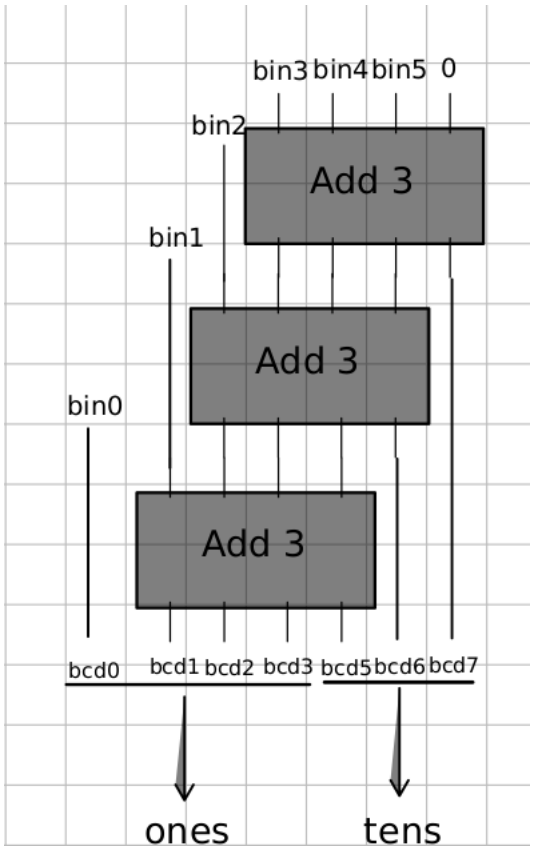

Figure 2: 6 Bit binary to BCD convertor

1. First 3-adder takes b_3, b_4, b_5 and 0 as input and its output is processed as follows
   (a) Carry - Carry is ignored because it is always going to be zero as the maximum value of input is going to be 7 and hence on adding 3 well get output of 10 which is well within the range of numbers 4 bits can hold.
   (b) MSB of sum - MSB of sum is redirected to output (d_6) because it will not be changed in any of the further steps and hence itll appear in output as it is.
   (c) Other bits of sum - Other bits of sum(S_0, S_1 and S_2) are redirected as input to the second adder.

2. Second 3-adder takes b_2 as input and three bits from the first 3-adder as input. This input corresponds to the shift step in Double Dabble algorithm. Its output is processed as follows

   (a) Carry - Carry is ignored because it is always going to be zero as the maximum value of input is going to be 7 and hence on adding 3 well get output of 10 which is well within the range of numbers 4 bits can hold.

   (b) MSB of sum - MSB of sum is redirected to output (d_5) because it will not be changed in any of the further steps and hence itll appear in output as it is.

   (c) Other bits of sum - Other bits of sum(S_0, S_1 and S_2) are redirected as input to the third adder.

3. Third 3-adder takes b1 as input and three bits from the first 3-adder as input. This input corresponds to the second shift step in the Double Dabble algorithm. Its output is processed as follows

   (a) Carry - Carry is ignored because it is always going to be zero as the maximum value of input is going to be 7 and hence on adding 3 well get output of 10 which is well within the range of numbers 4 bits can hold.

   (b) Sum - Sum is redirected to output ($d_2d_3d_4$)as no further step is involved.

4. Finally b_0 is directly redirected to d_0 as after this no shift or 3 addition is going to take place.

This completes the Double dabble algorithm for conversion of a 6 bit binary number to BCD.

3 7-bit binary to BCD convertor

Let the number be:-

$$b_0b_1b_2b_3b_4b_5b_6$$

and the BCD value be:-

$$d_1d_2d_3d_4d_5d_6d_7d_8d_9d_{10}d_{11}d_{12}$$

In the 7 bit binary to BCD converter, we used five units of 3-adders as explained below

1. First 3-adder takes b_4, b_5, b_6 and 0 as input and its output is processed as

   (a) Carry - Carry is ignored because it is always going to be zero as the maximum value of input is going to be 7 and hence on adding 3 well get output of 10 which is well within the range of numbers 4 bits can hold.

   (b) MSB of sum - MSB of sum is redirected to fifth adder (as third bit) which will carry out add-3 step in tens place if its value exceeds 5 before any shift.

   (c) Other bits of sum - Other bits of sum(S_0, S_1 and S_2) are redirected as input to the second adder.

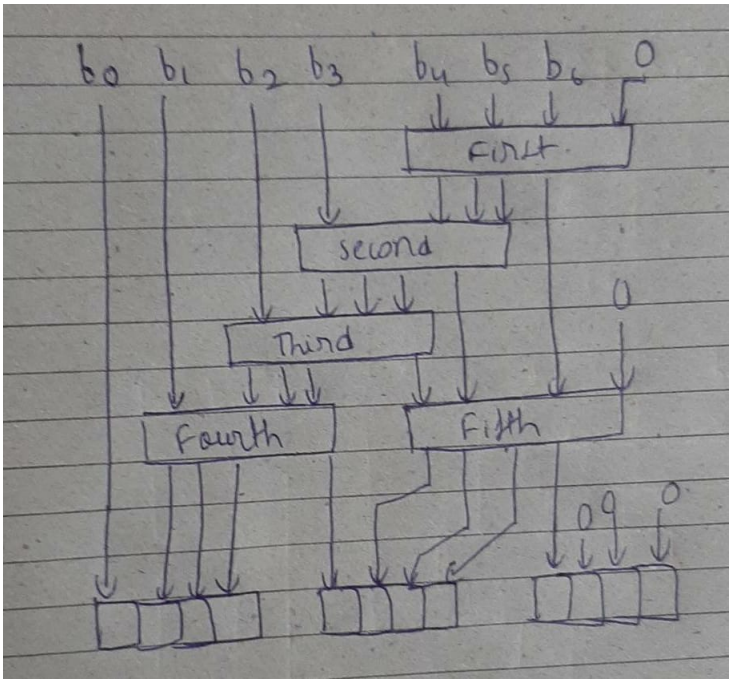

Figure 3: 7 Bit binary to BCD convertor

2. Second 3-adder takes b_3 as input and three bits from the first 3-adder as input. This input corresponds to the shift step in Double Dabble algorithm. Its output is processed as follows:-

   (a) Carry - Carry is ignored because it is always going to be zero as the maximum value of input is going to be 7 and hence on adding 3 well get output of 10 which is well within the range of numbers 4 bits can hold.

   (b) MSB of sum - MSB of sum is redirected to fifth adder (as second bit) which will carry out add-3 step in tens place if its value exceeds 5 before any shift.

   (c) Other bits of sum - Other bits of sum(S_0, S_1 and S_2) are redirected as input to the third adder.

3. Third 3-adder takes b_2 as input and three bits from the first 3-adder as input. This input corresponds to the second shift step in the Double Dabble algorithm. Its output is processed as follows:-

   (a) Carry - Carry is ignored because it is always going to be zero as the maximum value of input is going to be 7 and hence on adding 3 well get output of 10 which is well within the range of numbers 4 bits can hold.

   (b) MSB of sum - MSB of sum is redirected to fifth adder (as LSB bit) which will carry out an add-3 step in tens place if its value exceeds 5 before any shift.

   (c) Other bits of sum - Other bits of sum(S_0, S_1 and S_2) are redirected as input to the fourth adder.

4. Fourth 3-adder takes b_1 as input and three bits from the first 3-adder as input. This input corresponds to the third shift step in the Double Dabble algorithm. Its output is processed as follows:-
   (a) Carry - Carry is ignored because it is always going to be zero as the maximum value of input is going to be 7 and hence on adding 3 well get output of 10 which is well within the range of numbers 4 bits can hold.
   (b) Sum - Sum is redirected to output ($d_2d_3d_4$) as no further step is involved.
5. Fifth takes the MSBs of output of first, second and third adders along with 0 as input and its output is processed as follows:-
   (a) Carry - Carry is ignored because it is always going to be zero as the maximum value of input is going to be 7 and hence on adding 3 well get output of 10 which is well within the range of numbers 4 bits can hold.
   (b) Sum - Sum is redirected to output ($d_5d_6d_7d_9$)as no further step is involved.
6. Finally b_0 is directly redirected to d_0 as after this no shift or 3 addition is going to take place.

This completes the Double dabble algorithm for conversion of a 7 bit binary number to BCD.

4 Group Details

1. Kaushal Banthia - 19CS10039
2. Rohit Raj - 19CS10049
3. Animesh Jha - 19CS10070
4. Nisarg Upadhyaya - 19CS30031
5. Pranav Rajput - 19CS30036