Design and Implementation of a Generalized N-Digit Binary-To-Decimal Converter on an FPGA Seven-Segment Display Using Verilog Hdl

Asst. Lec. Asma Taha Saadoon
University of Baghdad/ Engineering College
Computer Engineering Department
asma_1683@yahoo.com

ABSTRACT

It is often needed to have circuits that can display the decimal representation of a binary number and specifically in this paper on a 7-segment display. In this paper a circuit that can display the decimal equivalent of an n-bit binary number is designed and it's behavior is described using Verilog Hardware Descriptive Language (HDL). This HDL program is then used to configure an FPGA to implement the designed circuit.

KEYWORDS: binary to decimal converter, FPGA, Verilog HDL, seven segment display, Cyclone II de1 board.

تصميم وتنفيذ محول من التمثيل الرقمي الثنائي الى التمثيل الرقمي العشري المكون من سلسلة رقمية طولها (N) على شاشة العرض ذات السبع قطع لحيز البوابات المرتبة بهيئة صفوف القابلة للبرمجة (FPGA) باستخدام لغة وصف الأجهزة فيرلوغ (verilog)

م.م. أسماء طه سعدون
جامعة بغداد/ كلية الهندسة
قسم هندسة الحاسبات

الخلاصة

غالبا ما تكون الحاجة الى دوائر تعرض العدد ممثل بالنظام العشري المكافئ الى ذلك العدد ممثل بالنظام الثنائي خصوصا في هذا البحث على شاشة العرض ذات السبع قطع. في هذا البحث تم تصميم دائرة لعرض التمثيل العشري لعدد ممثل بالنظام الثنائي طوله (n) وسلوك هذه الدائرة موصوف باستخدام لغة فيرلوغ (verilog) . واستخدم هذا الوصف بلغة فيرلوغ لتنفيذ الدائرة على ال FPGA.

الكلمات الرئيسية: محول من النظام العشري الى النظام الثنائي، حيز البوابات المرتبة بهيئة صفوف القابلة للبرمجة PGAF، لغة وصف الأجهزة فيرلوغ (verilog)، شاشة العرض ذات السبع قطع، لوحةCyclone II de1.

1. INTRODUCTION

In many electronic applications outputs are in binary form especially circuits designed using Hardware Descriptive Languages. Most of these applications require displaying the decimal equivalent of their outputs on for example a seven segment display. The design of a system that can be an interface between those outputs and a seven segment display is necessary, which is the aim of this work.

1.1 Related Work

Altera is a corporation that supplies programmable semiconductors, it provides a software tool called Quartus II to reconfigure these programmable devices. [Altera laboratory exercise, 2006] gives a basic idea of how to display the decimal equivalent of a 4-bit binary number on a seven segment display, its details are given in appendix A at the end of this paper. In addition [Wan-Fu H., 2011] has implemented a digital clock being displayed on a seven segment display, it differs with this paper in the objective but has some relevance about being both displayed on a seven segment display.

1.2 The Aim Of This Work

The aim of this work is to design a system that can display the decimal equivalent of any n-bit binary number on a seven segment display. This system is designed using Verilog HDL and then is implemented on an FPGA. In this work the idea in [Altera laboratory exercise, 2006] is developed to convert an n-bit binary number to its decimal equivalent. The complete system of this converter is explained in block form shown in **Fig. 1** and then each block is designed. Each block is explained by a truth table and a program code written in verilog HDL that describes its operation. The complete system is then written in verilog HDL and then downloaded on the programmable device (the FPGA) to implement it.

This paper is organized as follows, section 2 shows the block diagram of the system, and then each block in **Fig. 1** is explained in detail with its code written in verilog HDL. In section 3 the systems behavior is shown including its code in Verilog HDL. In section 4 hardware implementation and results are given. The paper is concluded in section 5.

2. THE PROPOSED SYSTEM BLOCK DIAGRAM

The complete design is shown in block form in **Fig. 1**. The block diagram is composed mainly of three circuits :

1. A circuit that compares between two numbers and produces an output accordingly, this circuit is called in this paper **the compare circuit**. It can be seen in **Fig. 1** that there is compare(1), compare(2),….,compare(N) this will be explained in the next sections.
2. A circuit that converts it's input to a value that is always between (0-9), and hence controls the seven segment display, this circuit is called in this paper **the convert circuit**. Since N-digits are assumed (N- seven segment displays) so there will be N-convert circuits.
3. The third circuit takes the output of the convert circuit and converts it to another code that will give the right display of the decimal digit on the seven segment display, this circuit is called in this paper the **seg7_display circuit**.

A general idea for the block diagram in **Fig. 1** is that the n-bit binary input (**in the bottom left**) enters the compare1 circuit. The compare1 circuit gives an output that together with the n-bit binary input enter the convert1 circuit. The output of the convert1 circuit then enters the seg7_display1 circuit. Now the latter controls the first seven segment display (HEX0) to give the required digit. The output of the compare1 circuit enters the compare2 circuit and gives an output that together with the input to the compare2 circuit enter the convert2 circuit. The output of the convert2 circuit enter the seg7_display2 circuit and controls the second seven segment display (HEX1). This operation will be repeated for all digits until the full decimal number is obtained. In the next sections the operation of each circuit is explained in detail.

2.1 The Compare Circuit

The compare circuit has 1 input and 1 output as shown in **Fig. 1**. It checks whether its input is between [0,9]* if not then it checks if it's between [10,19] if not then it checks if it's between [20,29] and so on, and gives a unique output for each case. **Table 1** shows the truth table of this circuit where its input has n bits and its output has (n-3) bits. For example If a 5-bit input is applied to the compare circuit (as in **Table 1**) then the output has 2 bits. As n increases the number of output bits also increases, for example if n = 6 then the decimal output is between [0,63]. In this case the compare circuits output has three bits to show the seven different cases ([0-9], [10-19], [20-29],….., [60-63]). The verilog HDL code describing the compare circuit is shown in **Program code(1)**.

2.2 The Convert Circuit

The convert circuit has two inputs and one output, one of the inputs is the n-bit binary number and the other is the output of the compare circuit. **Table 2** shows the truth table for this circuit also for a 5-bit input as in the compare circuit. **Program code(2)** shows the code describing the convert circuits behavior written in verilog HDL language. The convert circuit converts the input to a value that is always between 0 and 9, that will be displayed on the seven segment via the **seg7_display** circuit.

In **Table 2** if the n-bit input (x) is 00000, 01010, 10100, or 11110 the output (Y) for all of these inputs is 00000, the decimal value of this output (y) is 0 which equals the first digit D_0 (first seven segment display). The second digit D_1 for the same mentioned inputs also from **Table 2** equals the decimal value of the second input (z) that is the output of the compare circuit, so in order to display D_1 then z is applied to another compare circuit (compare2) and its output will be applied to another convert circuit (convert2) that will display D_1 on the seven segment display (HEX1) via the second seg7_display circuit (seg7_display2). The input z can be thought of as a control variable, let **init** = 10 (decimal), it can be seen from **Table 2** that if z=00 then y = x-0*init, else if z =01 then y = x-init*1, else if z = 10 then y = x-init*2 , else if z = 11 then y = x-init*3, this is designed using verilog language by a while loop and making z the control variable as shown in Program code(2).

2.3 Seg7_display

The seg7_display has one n-bit input and a one 7-bit output. The input to this circuit is the output of the convert circuit which when applied to the seg7_display circuit will be decoded to another code. The seg7_display is designed in verilog HDL by [Ciletti. M. D., 2005] and the same design is used in this paper. The seven segment display shown in **Fig. 1** (HEXi, where i = 0, 1, 2, …,N-1) has 7 light-emitting diodes which illuminate when a low voltage is applied to them. So by illuminating the right diodes the required decimal value is displayed. In **Program code(3)** the code in verilog HDL is given [Ciletti. M. D., 2005].

* **The [ ] means a closed interval.**

Table 1 The compare circuit truth table

n-bit input (5-bit) $b_4\ b_3\ b_2\ b_1\ b_0$	(n-3) output (2-bit) $C_1\ C_0$	Decimal equivalent $D_1\ D_0$
00000	00	00
00001	00	01
00010	00	02
00011	00	03
00100	00	04
.	.	.
01001	**00**	**09**
01010	01	10
01011	01	11
.	.	.
01111	01	15
.	.	.
.	.	.
10011	**01**	**19**
10100	10	20
.	.	.
.	.	.
11101	**10**	**29**
11110	11	30
11111	11	31

```
module compare (b,c);
parameter no_bit = n;
parameter comt_sig = (n-3);
parameter comp = 9;
parameter add = 10;
input [no_bit-1:0] b;
output [comt_sig -1:0] c;
reg [ comt_sig -1:0] c;
integer count;
always @ (b) begin
count = 0;
while ( b > (comp + count*add)) begin
count = count +1;
end
c = count;
end
endmodule
```

Program code(1) The compare circuit code in Verilog HDL

```
module convert (x,z,y);
parameter  no_bit1 = n;
parameter  comt1_sig = (n-3);
parameter  init = 10;
input [no_bit1 -1:0] x;
input [comt1_sig -1:0]z;
output [no_bit1-1:0] y;
reg [no_bit1-1:0] y;
integer q;
always @ (x , z) begin
q = 0;
while (q < (2**comt1_sig)) begin  // **
means to the power
if (z == q) begin
y <= x - q*init;
end
q = q + 1;
end
end
endmodule
```

Program code(2) The convert circuit code in Verilog HDL

Table (2) The convert circuit truth table

n-bit input (5-bit) $x_4\ x_3\ x_2\ x_1\ x_0$	**(n-3) input (2-bit)** $z_1\ z_0$	**n-bit output** $y_4\ y_3\ y_2 y_1\ y_0$	**Decimal equivalent** $D_1\ D_0$
00000	**00**	**00000**	**00**
00001	00	00001	01
00010	00	00010	02
00011	00	00011	03
00100	00	00100	04
00101	00	00101	05
.	.	.	.
01001	**00**	**01001**	**09**
01010	**01**	**00000**	**10**
01011	01	00001	11
.	.	.	.
01111	01	00101	15
.	.	.	.
10011	**01**	**01001**	**19**
10100	**10**	**00000**	**20**
.	.	.	.
11101	**10**	**01001**	**29**
11110	**11**	**00000**	**30**
11111	11	00001	31

```
module seg7_display (m,n);
input [n-1:0] m;
output [6:0] n;
reg [6:0] n;
parameter blank = 7'b 1111111;
parameter zero = 7'b 1000000;
parameter one = 7'b 1111001;
parameter two = 7'b 0100100;
parameter three = 7'b 0110000;
parameter four = 7'b 0011001;
parameter five = 7'b 0010010;
parameter six = 7'b 0000010;
parameter seven = 7'b 1111000;
parameter eight = 7'b 0000000;
parameter nine = 7'b 0010000;
always @ (m) begin
case (m)
0: n <= zero;
1: n <= one;
2: n <= two;
3: n <= three;
4: n <= four;
5: n <= five;
6: n <= six;
7: n <= seven;
8: n <= eight;
9: n <= nine;
default: n <= blank;
endcase
end
endmodule
```

Program code(3) The seg7_display circuit code in Verilog HDL.

3. THE PROPOSED SYSTEM HDL DESIGN

The complete design in verilog HDL for the block diagram of **Fig. 1** is shown in **Program code(4)**. This code shows four 7-segment displays (HEX0, HEX1, HEX2, HEX3), one for each decimal digit (it can easily be generalized to N digits).

To understand the binary-to-decimal converter circuit, take n=5. Observe **Fig.1**, **Table 1** and **Table 2** it can be seen that:

1. If a binary input with n= 5, is applied to compare1, output **(c)** is as in **Table 1**.
2. Now this same binary input is applied together with **(c)** to the convert1 circuit, which gives **(y)** as in **Table 2**. Output y is always between (0-9) so if for example:
   The binary input is 15 (01111), the first digit is 5 and the second is 1. In this case c =01, which means that the binary input is between [10-19]. At the same time y = binary input – 10, y = 15-10=5. Y is applied to the seg7_display1 that gives an output of (0010010) to the first 7-segment display (HEX0). This bit pattern illuminates LED0, LED2, LED3, LED5, and LED6 of the (HEX0) 7-segment display which in return displays a 5.
3. Now to display the second digit on the second seven segment display HEX1, observe the output (z) in **Table 1** it can be seen that it corresponds to the second digit. So (z) is applied to compare2 and an output z' is obtained. z together with z' is applied to convert2 and gives an output y'. y' is applied to the seg7_display2 which gives the binary bit pattern that illuminates the correct LEDs that shows the decimal value (in this example it shows a **1**).
4. The same is repeated as is shown in the block diagram of **Fig. 1** to display all digits on all displays.

4. HARDWARE IMPLEMENTATION AND RESULTS

The convert_b_to_d (**program code(4)**) is written in verilog HDL using Quartus II software tool version 7.2 [Altera software installation manual, 2011]. **Program code(4)** is downloaded on an FPGA (Cyclone II EP2C20F484C7) [Cyclone II, User Guide], and the flow summary of the compilation report is shown in **Table 3 and Table 4** (this flow summary is if the number of input binary bits is 10, which is the maximum number for this FPGA kit). Table 3 and 4 show that only 206 logic elements (1%) are used, which means that a very small part of the FPGA kit is reconfigured to implement this system.

In **Fig. 2** the cyclone II EP2C20F484C7 starter development board is shown with the seven segment display and the toggle switches (SW) [Cyclone II, Reference manual]. In the figure the toggle switches (SW (SW_0, SW_1, ….SW_9)) represent the binary input and the seven segment display represent their decimal equivalent. When the toggle switches position is down (up)[*] then the input is logic **"0"** (**"1"**).

[*] Down (up) is the bottom (top) of the page.

```
module convert_b_to_d (sw,
hex0,hex1,hex2,hex3);
parameter bit_input = n;
parameter comp_out = n-3;
input [bit_input -1:0] sw;
output [6:0] Hex0, Hex1, Hex2, Hex3;
// array of wires ( 4 is the number of
digits) in general N.
wire [bit_input-1:0] kwire [4:0];
wire [comp_out-1:0] owire [3:0];
wire [bit_input-1:0] kowire [3:0];
  wire [6:0] hex [3:0];
// module loop using generate
genvar i;
generate for ( i = 0; i <4; i = i+1 )
  begin: inst
assign kwire[0]=sw;
compare compare_i (kwire[i], kowire[i] );
convert convert_i (kwire[i],
kowire[i],owire[i]);
seg7_display seg7_display_i
(owire[i],hex[i]);
assign kwire[i+1]=kowire[i];
    end
assign hex0=hex[0];
assign hex1=hex[1];
assign hex2=hex[2];
assign hex3=hex[3];
  endgenerate
endmodule
```

Program code(4) The convert_b_to_d system code in Verilog HDL

Table (3) The compilation flow

Compilation Flow Status	**Successful**
Quartus II Version	7.2 Build 151 09/26/2007 SJ
Revision Name	convert_b_to_d
Top-level Entity Name	convert_b_to_d
Family	Cyclone II
Device	EP2C20F484C7
Timing Models	Final
Met timing requirements	Yes
Total logic elements	206 / 18,752 (1 %)
Total combinational functions	206 / 18,752 (1 %)
Dedicated logic registers	0 / 18,752 (0 %)
Total registers	0
Total pins	38 / 315 (12 %)
Total virtual pins	0
Total memory bits	0 / 239,616 (0 %)
Embedded Multiplier 9-bit elements	0 / 52 (0 %)
Total PLLs	0 / 4 (0 %)

Table (4) The Analysis and synthesis

Analysis & Synthesis Status	**Successful**
Quartus II Version	7.2 Build 151 09/26/2007 SJ
Revision Name	convert_b_to_d
Top-level Entity Name	convert_b_to_d
Family	Cyclone II
Total logic elements	206
Total combinational functions	206
Dedicated logic registers	0
Total registers	0
Total pins	38
Total virtual pins	0
Total memory bits	0
Embedded Multiplier 9-bit elements	0
Total PLLs	0

5. CONCLUSION

In this paper a binary-to-decimal converter is designed and implemented on an FPGA. This design can be used for any number of input binary bits, which can be made directly by changing only the number of input bits in the code and increasing the number of digits that will display the decimal number. Table 3, and 4 show that a small part of the FPGA is reconfigured to implement this system, and since this system is used mainly as an interface between any electronic circuit implemented on an FPGA and a seven segment display then the small size is a good feature.

REFERENCES

Altera corporation, Laboratory Exercise 2" Numbers and Displays" 2006, ftp://ftp.altera.com/up/pub/Laboratory_Exercises/DE1/Digital_Logic/Verilog/lab2_Verilog.pdf.

Altera Software Installation and Licensing Manual, 101 Innovation Drive, San Jose, CA 95134, November 2011 Altera Corporation.

Ciletti. M. D. , "Advanced Digital Design with the Verilog HDL", Prentice-Hall of India, New Delhi, 2005.

Cyclone II FPGA Starter Development Board, Reference manual, Altera San Jose, CA 95134, October 2006.

Cyclone II FPGA Starter Development Kit, User Guide, Altera San Jose, CA 95134, October 2006.

Wan-Fu H., "The design of a 6-digit digital clock with a four-digit seven-segment display module", IEEE Electrical and Control Engineering, International conference Page(s): 2656 – 2659, 2011.

LIST OF ABBREVIATIONS:

FPGA: Field Programmable Gate Array.
HDL: Hardware Descriptive Language.
LED: Light Emitting Diode.

APPENDIX A

You are to design a circuit that converts a four-bit binary number *V* = *v*3*v*2*v*1*v*0 into its two-digit decimal equivalent *D* = *d*1*d*0. **Table A1** shows the required output values. A partial design of this circuit is given in **Fig. A1**. It includes a comparator that checks when the value of *V* is greater than 9, and uses the output of this comparator in the control of the 7-segment displays. You are to complete the design of this circuit by creating a Verilog module which includes the comparator, multiplexers, and circuit *A* (do not include circuit *B* or the 7-segment decoder at this point). Your Verilog module should have the four-bit input *V* , the four-bit output *M* and the output *z*.

Table A1 Binary-to-decimal conversion values

Binary value	Decimal Digit
0000	0 0
0001	0 1
0010	0 2
. . .	. . .
1001	0 9
1010	1 0
1011	1 1
1100	1 2
1101	1 3
1110	1 4
1111	1 5

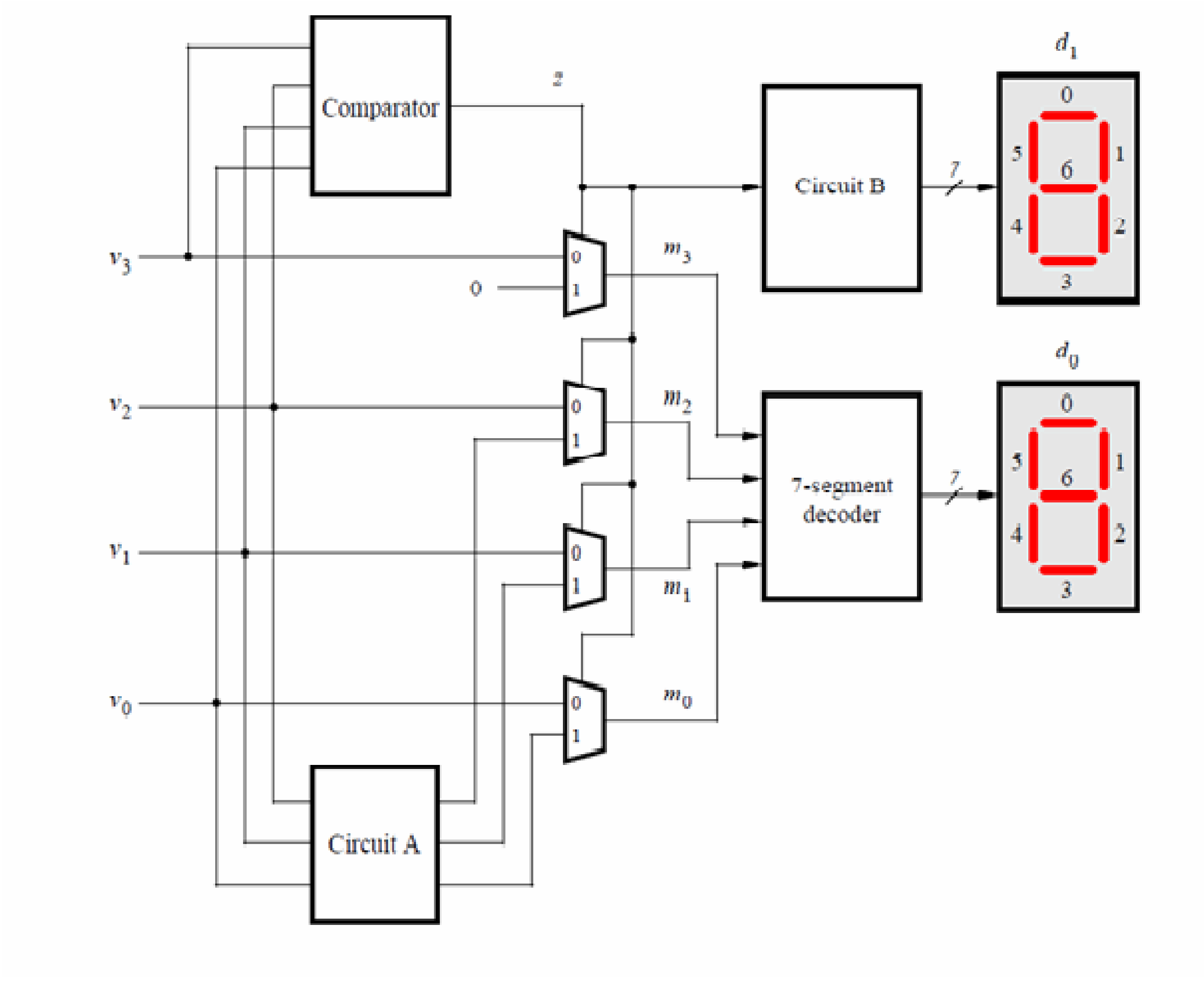

Fig. A1 Partial design of the binary-to-decimal conversion circuit.

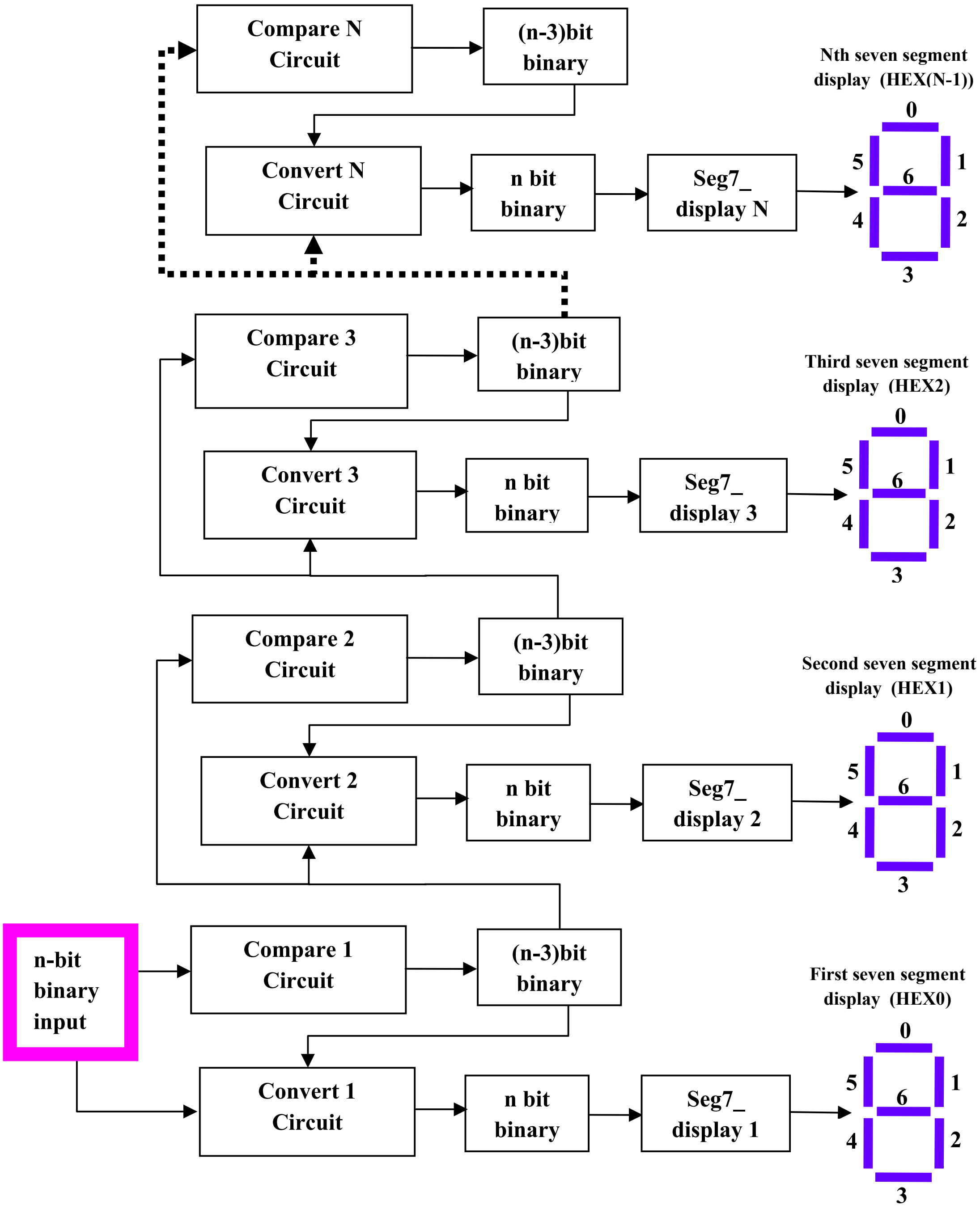

Fig. 1 The binary-to-decimal converter block diagram

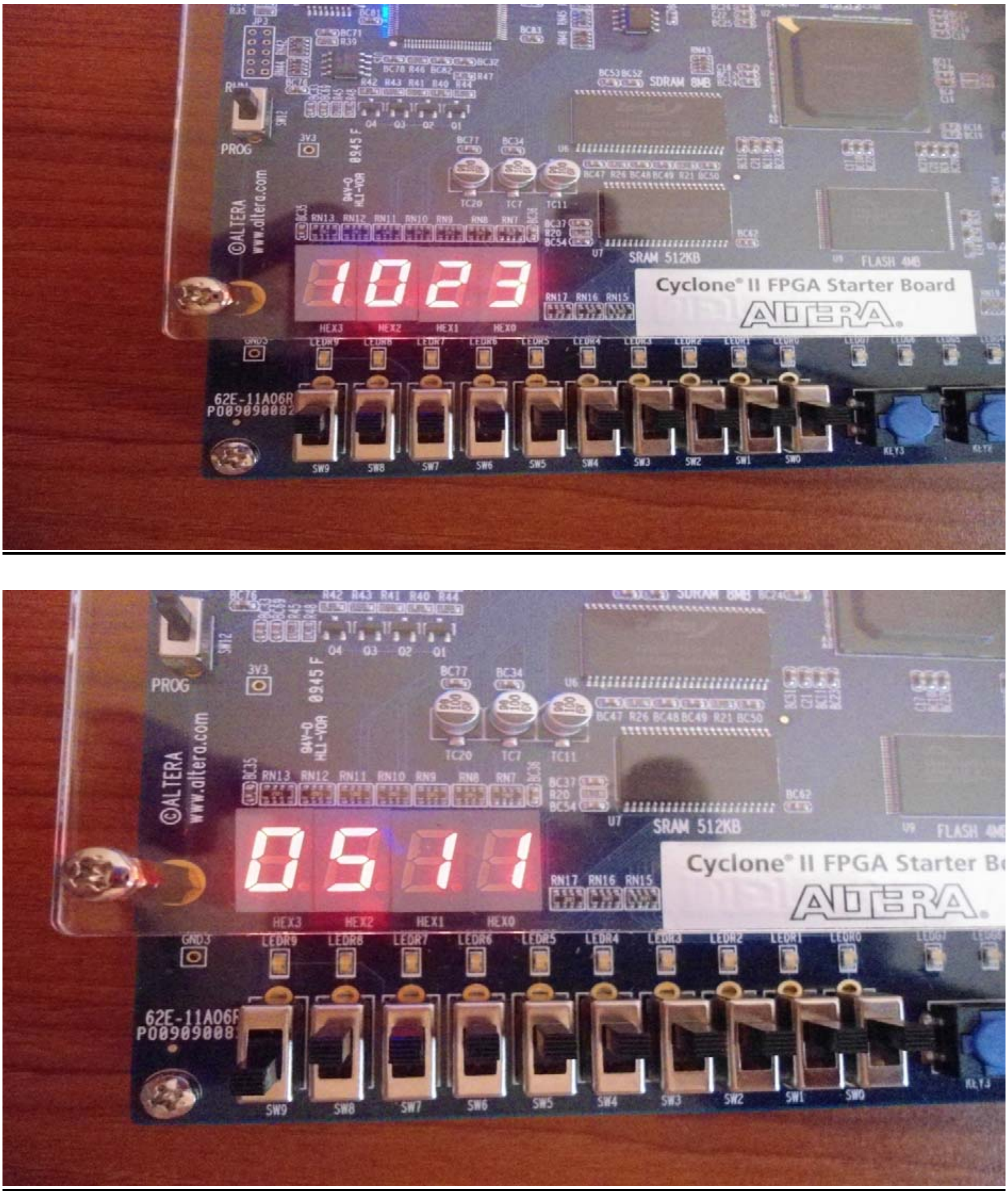

Fig. 2 Cyclone II EP2C20F484C7 FPGA seven segment display