

## Комментарии и советы по выполнению работы Lab3\_2

1. ВОПРОС: «В лабораторной работе требуется в качестве выхода модуля указать двумерный массив (7 разрядов десятков и 7 разрядов единиц). Как я понял сделать это на Verilog невозможно (в SystemVerilog такая возможность есть). Будет ли критично заменить двумерный массив на обычный вектор. Так как в любом случае при назначении портов мы сможем указать разряды вектора. Или возможно есть способ сделать это на Verilog? (output [6:0] hex\_int [1:0])»

ОТВЕТ: Да, нужно указать как два вектора. Например, так, как показано ниже.

```
1  module Lab3_2 #(parameter div = 4) (  
2      input CLK,  
3      input RST,  
4      input ENA,  
5      input DIR,  
6      output [6:0]HEX_1,  
7      output [6:0]HEX_0);  
8  
9      wire int_rst;  
10     wire int_ena;  
11     wire int_dir;  
12     wire cout;  
13     wire cnt_ena;  
14     wire [5:0]cnt_val;  
15     wire [6:0]int_hex_1;  
16     wire [6:0]int_hex_0;  
17  
18     SYNCin SYNCin_RST (  
19         .CLK      (CLK      ),  
20         .D_in      (RST      ),  
21         .D_out      (int_rst  )  
22     );  
23     SYNCin SYNCin_ENA (  
24         .CLK      (CLK      ),  
25         .D_in      (ENA      ),  
26         .D_out      (int_ena  )  
27     );  
28     SYNCin SYNCin_DIR (  
29         .CLK      (CLK      ),  
30         .D_in      (DIR      ),  
31         .D_out      (int_dir  )  
32     );  
33     CNT_DIV #(div) CNT_DIV_inst (  
34         .CLK      (CLK      ),  
35         .RST      (int_rst  ),  
36         .ENA      (int_ena  ),  
37         .C_out      (cout    )  
38     );  
39     assign cnt_ena = int_ena & cout;  
40     CNT #(6'd23) CNT_inst (  
41         .CLK      (CLK      ),  
42         .RST      (int_rst  ),  
43         .ENA      (cnt_ena  ),  
44         .DIR      (int_dir  ),  
45         .CNT_out   (cnt_val  )  
46     );  
47     CODER CODER_inst (  
48         .address (cnt_val    ),  
49         .clock   (CLK        ),  
50         .q       ({int_hex_1,int_hex_0} )  
51     );  
52     SYNCout #(14) SYNCout_inst (  
53         .CLK      (CLK      ),  
54         .RST      (int_rst  ),  
55         .ENA      (int_ena  ),  
56         .D_in      ({int_hex_1,int_hex_0} ),  
57         .D_out      ({HEX_1, HEX_0} )  
58     );  
59 endmodule
```

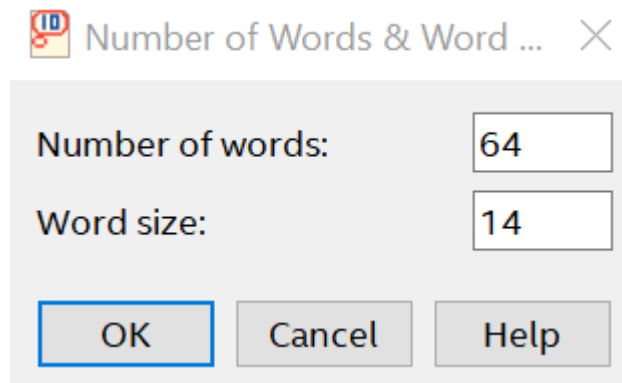
2. ВОПРОС: «Также возникли проблемы с созданием hex файла для ROM-1 компонента. В целом я разобрался с Intel hex форматом, но возникло чувство, что он должен был быть приложен к работе. Найти простой способ генератора формата не удалось, а расчет контрольной суммы для 32 значений оказался достаточно трудоемким занятием.»

ОТВЕТ: Таблицу (на все 64 значения – разрядность счетчика 6 бит) легко построить в Excel (либо написать формирование текстовой таблицы на Python или Си или....). Пример таблицы в Excel приведен ниже (делается минут за 5 путем ручного копирования, если данные рассматривать как два вектора по 7 бит в каждом.)

адрес	данные
0	10000001000000
1	10000001111001
2	10000000100100
3	10000000110000
4	10000000011001
5	10000000010010
6	10000000000010
7	10000001111000
8	10000000000000
9	10000000010000
10	11110011000000
11	11110011111001
12	11110010100100
13	11110010110000
14	11110010011001
15	11110010010010
16	11110010000010
17	11110011111000
18	11110010000000
19	11110010010000
20	01001001000000
21	01001001111001
22	01001000100100
23	01001000110000
24	01001000011001
25	01001000010010
26	01001000000010
27	01001001111000
28	01001000000000
29	01001000010000
30	01001001000000
31	01001001111001
32	01001000100100
33	01001000110000
34	01001000011001
35	01001000010010
36	01001000000010
37	01001001111000
38	01001000000000
39	01001000010000
40	00110011000000
41	00110011111001
42	00110010100100
43	00110010110000
44	00110010011001
45	00110010010010
46	00110010000010
47	00110011111000
48	00110010000000
49	00110010010000
50	00100101000000
51	00100101111001
52	00100100100100
53	00100100110000
54	00100100011001
55	00100100010010
56	00100100000010
57	00100101111000
58	00100100000000
59	00100100010000
60	00000101000000
61	00000101111001
62	00000100100100
63	00000100110000

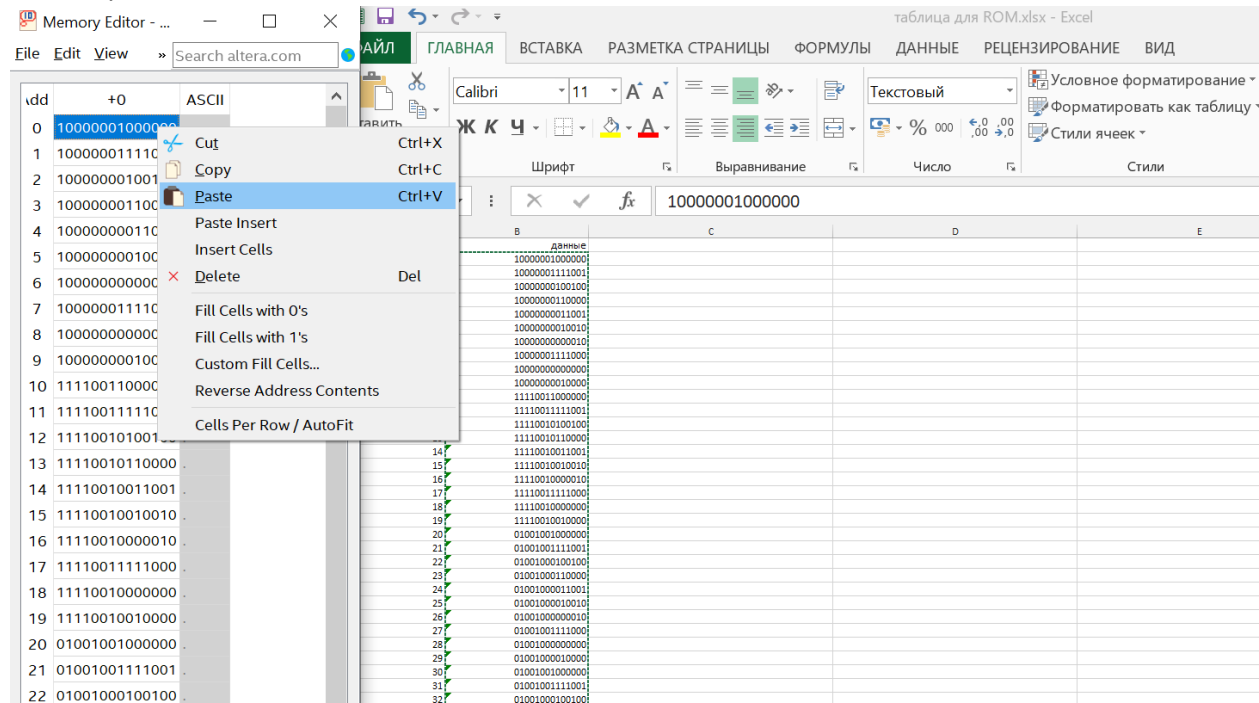
С HEX форматом файла разбираться было не нужно. Достаточно создать HEX (MIF) файл в пакете Quartus: **File=>New=>Memory Initialization File** (HEX или MIF на данном этапе можно выбрать любой).

Затем задаем число слов 64, разрядность слова 14:



Затем выделяем все ячейки данных в excel таблице.

Щелкаем правой клавишей в нулевой ячейке нашей таблицы (HEX или MIF) и выбираем команду Paste



Все данные будут перенесены.

Далее сохраняем наш HEX (MIF) файл с расширением .hex и получаем искомый файл в формате hex

```
1 :0200000020409E
2 :02000100207964
3 :02000200202488
4 :020003002030AB
5 :020004002019C1
6 :020005002012C7
7 :020006002002D6
8 :0200070020785F
9 :020008002000D6
10 :020009002010C5
11 :02000A003CC0F8
12 :02000B003CF9BE
13 :02000C003CA412
14 :02000D003CB005
15 :02000E003C991B
16 :02000F003C9221
17 :020010003C8230
18 :020011003CF8B9
19 :020012003C8030
20 :020013003C901F
21 :02001400124098
22 :0200150012795E
23 :020016001224B2
24 :020017001230A5
25 :020018001219BB
26 :020019001212C1
27 :02001A001202D0
28 :02001B00127859
29 :02001C001200D0
30 :02001D001210BF
31 :02001E0012408E
32 :02001F00127954
33 :020020001224A8
34 :0200210012309B
35 :020022001219B1
36 :020023001212B7
37 :020024001202C6
38 :0200250012784F
39 :020026001200C6
40 :020027001210B5
41 :020028000CC00A
42 :020029000CF9D0
43 :02002A000CA424
44 :02002B000CB017
45 :02002C000C992D
46 :02002D000C9233
47 :02002E000C8242
48 :02002F000CF8CB
49 :020030000C8042
50 :020031000C9031
51 :02003200094083
52 :02003300097949
53 :0200340009249D
54 :02003500093090
55 :020036000919A6
56 :020037000912AC
57 :020038000902BB
58 :02003900097844
59 :02003A000900BB
60 :02003B000910AA
61 :02003C00014081
62 :02003D00017947
63 :02003E0001249B
64 :02003F0001308E
65 :00000001FF
```

3. ПОДСКАЗКА: Перед полной компиляцией проекта (db\_Lab3\_2) установите опцию:

**Assignment=>Device=>Device and Pin Option => Configuration**

