VILNIUS GEDIMINAS
TECHNICAL UNIVERSITY
FACULTY OF ELECTRONICS
DEPARTMENT OF ELECTRONIC SYSTEMS

Variable frequency signal generation

Embedded Systems
Intermediate Exam 1

Done by: EKSfmu-16 gr. st. Arūnas Butkus
Checked by: doc. dr. Andrius Ušinskas

Vilnius, 2017

Work goal: using DE0 development board generate periodical signal showing frequency using HEX displays and allowing to change said frequency with switches.

Algorithms used in tasks.

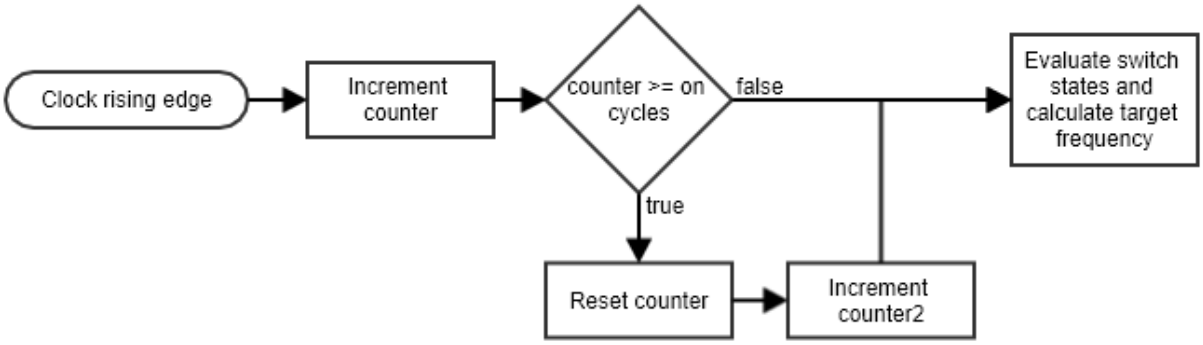


Fig. 1. "Algorithm" for signal generation.

Results analysis.

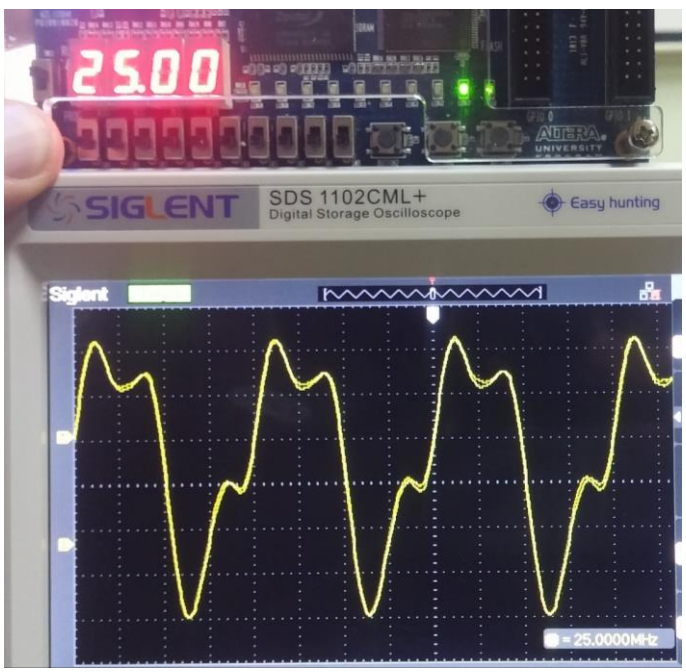


Fig. 2. Signal generation without any frequency division.

All in all it is simple operation. First on each clock rising edge a counter is increased and that value is tested against a threshold value *x* and, if counter is greater than *x*, then it is reset and another counter, *y*, is increased. Threshold value *x* is calculated from the switches – switches representing a binary number. 10 switches give *x* value ranging from 0 to 1023. In practice, *x* is on or off cycle count, since the second counter is used to determine the output state – output high when counter even, output low when counter odd. This means that it is effectively a frequency divider. In same cycle frequency for display is also determined, by a simple equation. HEX

displays are directly tied to that frequency value. Examples of operation are shown in Figure 3 and can also be viewed on YouTube at https://youtu.be/Wulkias8waE

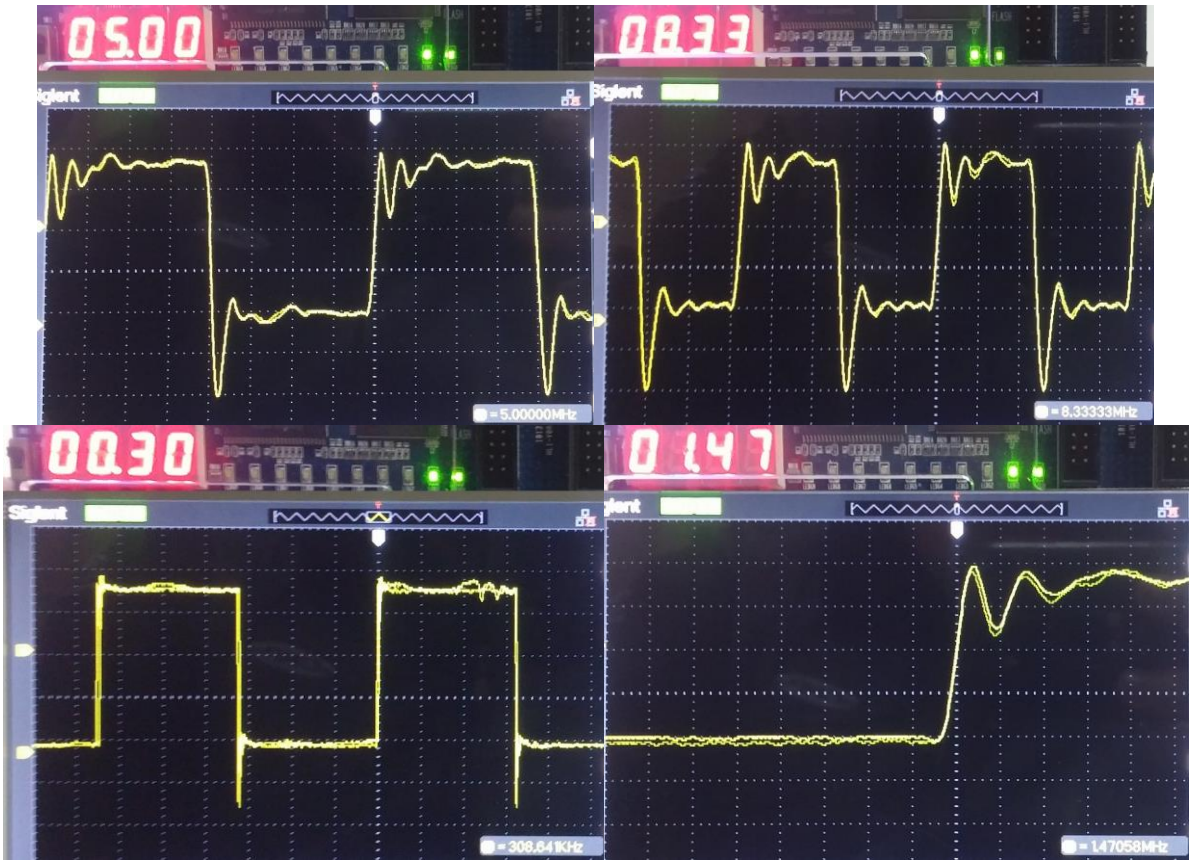


Fig. 3. Few examples of signal generation using the divider.

Conclusions.

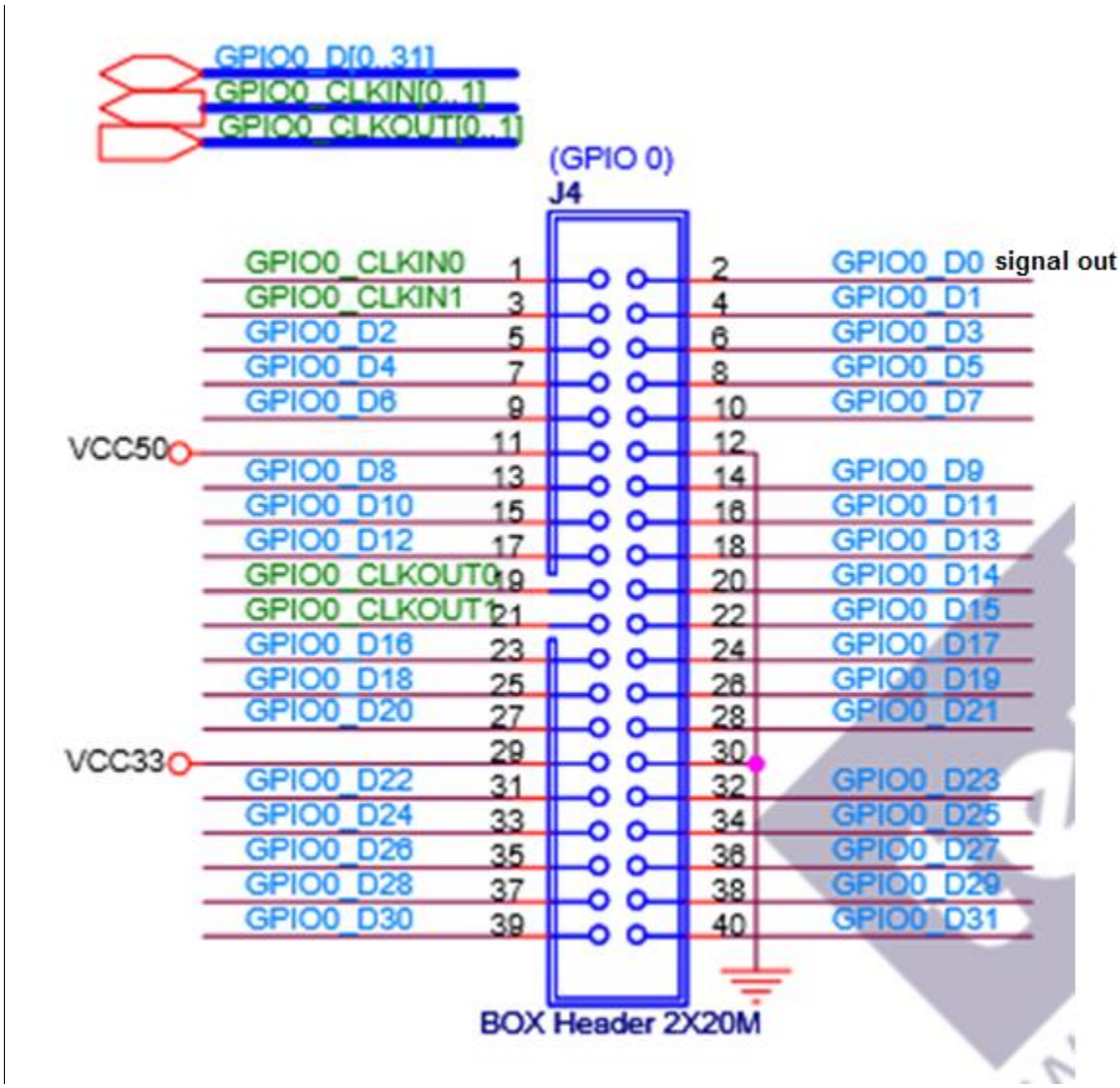
Well, it works and is quite precise. The 4 HEX displays can't provide much precision when indicating target frequency in some cases, but does point to the right direction. By the way the signal is a square wave with maximum frequency of 25 MHz. And in this particular setup minimum frequency is 48 kHz.

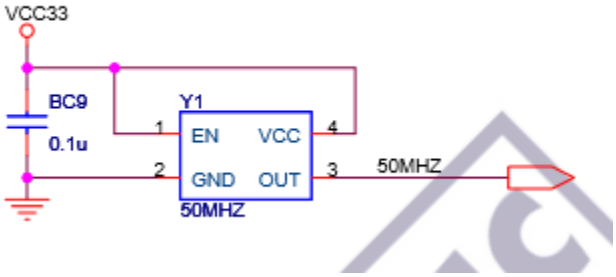
Source code.

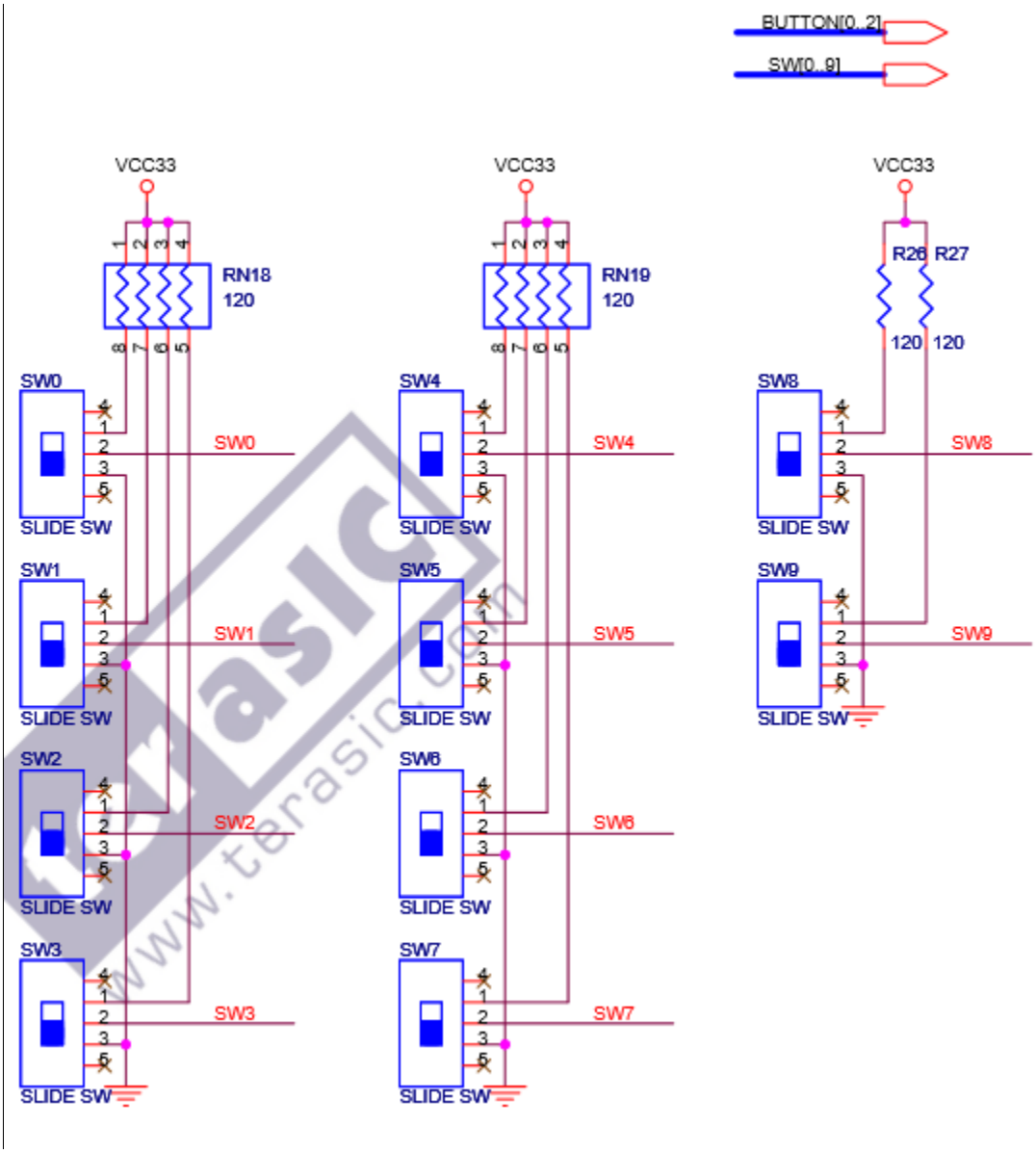
```
001 library ieee;
002 use ieee.std_logic_1164.all;
003 use ieee.numeric_std.all;
004
005 entity IESigGen is
006   Port (
007         CLOCK 50 : IN STD LOGIC;
008         SW : IN STD_LOGIC_VECTOR(9 DOWNTO 0);
009         KEY : IN STD_LOGIC_VECTOR(2 DOWNTO 0);
010         HEX0 : OUT STD_LOGIC_VECTOR(7 DOWNTO 0);
011         HEX1 : OUT STD_LOGIC_VECTOR(7 DOWNTO 0);
012         HEX2 : OUT STD LOGIC VECTOR(7 DOWNTO 0);
013         HEX3 : OUT STD_LOGIC_VECTOR(7 DOWNTO 0);
014         LEDG : OUT STD_LOGIC_VECTOR(9 downto 0);
015         GPIO_0 : OUT STD_LOGIC_VECTOR(2 DOWNTO 0)
016   );
017 end IESigGen;
018
019 architecture Whatever of IESigGen is
020
021 signal timer_count    : INTEGER := 0;
022 signal counter        : INTEGER := 0;
023 SIGNAL FREQUENCY      : INTEGER := 0;
024 signal INTERVAL       : INTEGER := 0;
025
026 begin
027
028   WITH FREQUENCY mod 10 SELECT HEX0 <=
029         "11000000" WHEN 0,
030         "11111001" WHEN 1,
031         "10100100" WHEN 2,
032         "10110000" WHEN 3,
033         "10011001" WHEN 4,
034         "10010010" WHEN 5,
035         "10000010" WHEN 6,
036         "11111000" WHEN 7,
037         "10000000" WHEN 8,
038         "10010000" WHEN 9,
039         "01111111" WHEN OTHERS;
040
041   WITH (FREQUENCY / 10) mod 10 SELECT HEX1 <=
042         "11000000" WHEN 0,
043         "11111001" WHEN 1,
044         "10100100" WHEN 2,
045         "10110000" WHEN 3,
046         "10011001" WHEN 4,
047         "10010010" WHEN 5,
048         "10000010" WHEN 6,
049         "11111000" WHEN 7,
050         "10000000" WHEN 8,
051         "10010000" WHEN 9,
052         "01111111" WHEN OTHERS;
053
054   WITH (FREQUENCY / 100) mod 10 SELECT HEX2 <=
055         "01000000" WHEN 0,
056         "01111001" WHEN 1,
057         "00100100" WHEN 2,
058         "00110000" WHEN 3,
059         "00011001" WHEN 4,
060         "00010010" WHEN 5,
```

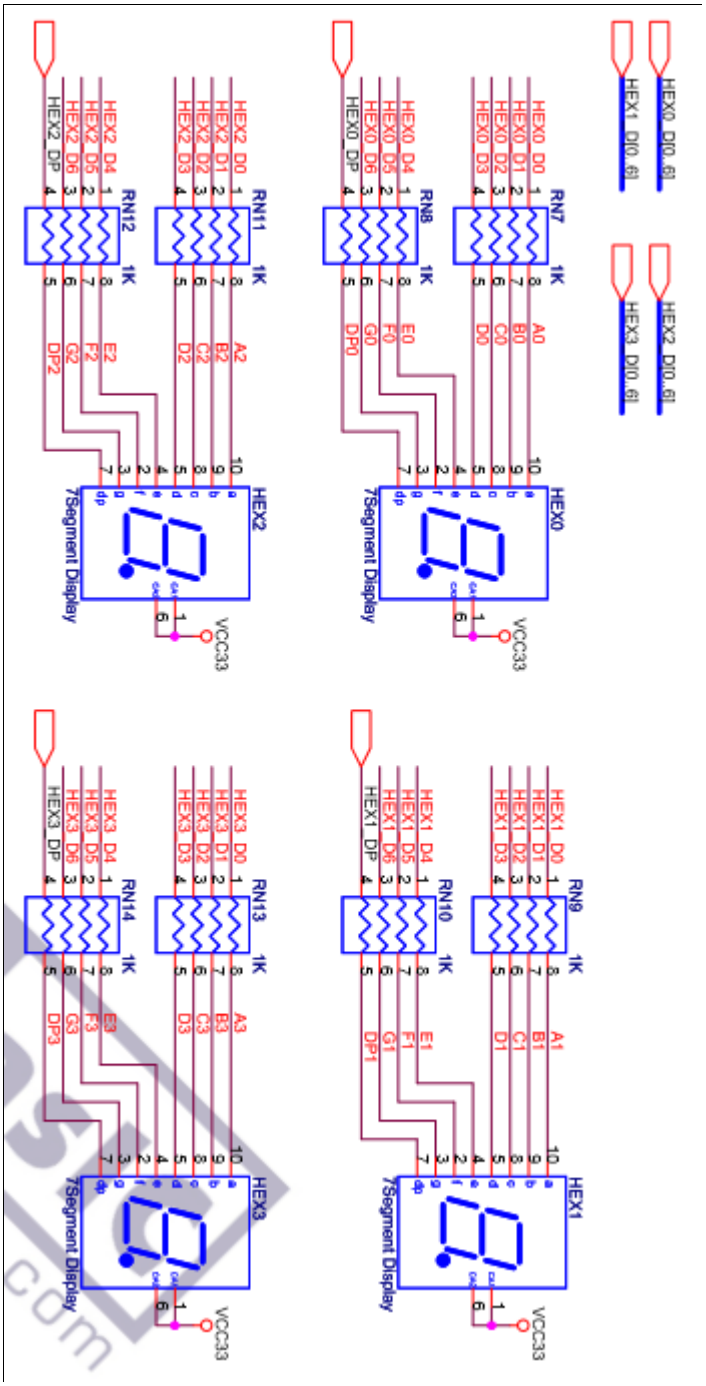
```
061         "00000010" WHEN 6,
062         "01111000" WHEN 7,
063         "00000000" WHEN 8,
064         "00010000" WHEN 9,
065         "01111111" WHEN OTHERS;
066
067   WITH (FREQUENCY / 1000) mod 10 SELECT HEX3 <=
068         "11000000" WHEN 0,
069         "11111001" WHEN 1,
070         "10100100" WHEN 2,
071         "10110000" WHEN 3,
072         "10011001" WHEN 4,
073         "10010010" WHEN 5,
074         "10000010" WHEN 6,
075         "11111000" WHEN 7,
076         "10000000" WHEN 8,
077         "10010000" WHEN 9,
078         "01111111" WHEN OTHERS;
079
080   WITH COUNTER MOD 2 SELECT GPIO_0 <=
081         "001" WHEN 0,
082         "000" WHEN OTHERS;
083
084   process(CLOCK_50)
085   begin
086         if rising_edge(CLOCK_50) then
087               timer_count <= timer_count + 1;
088               if (timer_count >= INTERVAL) then
089                     counter <= counter + 1;
090                     timer_count <= 0;
091               end if;
092
093               INTERVAL <= TO_INTEGER(UNSIGNED(SW));
094               FREQUENCY <= 5000/((INTERVAL+1)*2);
095
096         end if;
097   end process;
098
099 end Whatever;
```

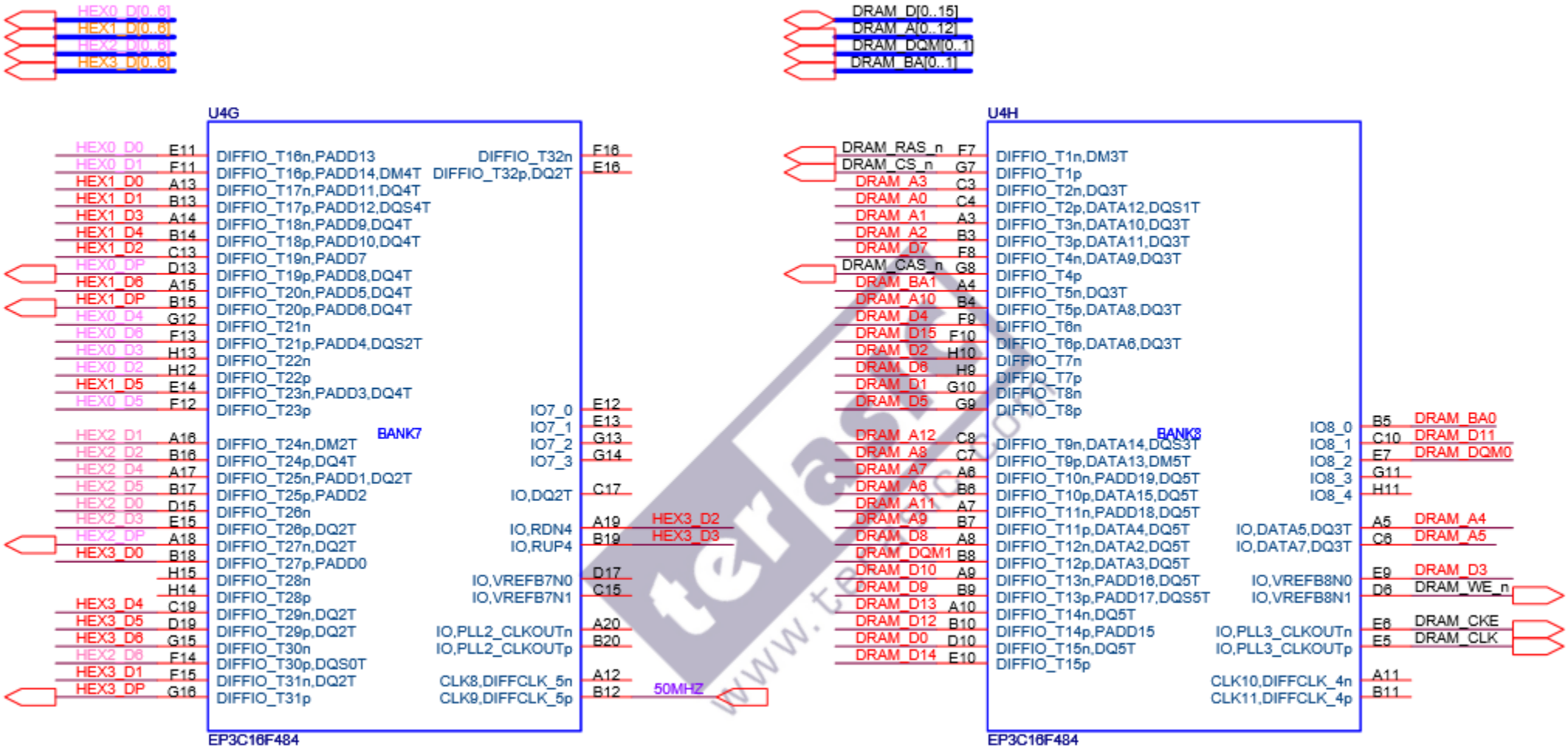
Electrical schemes:

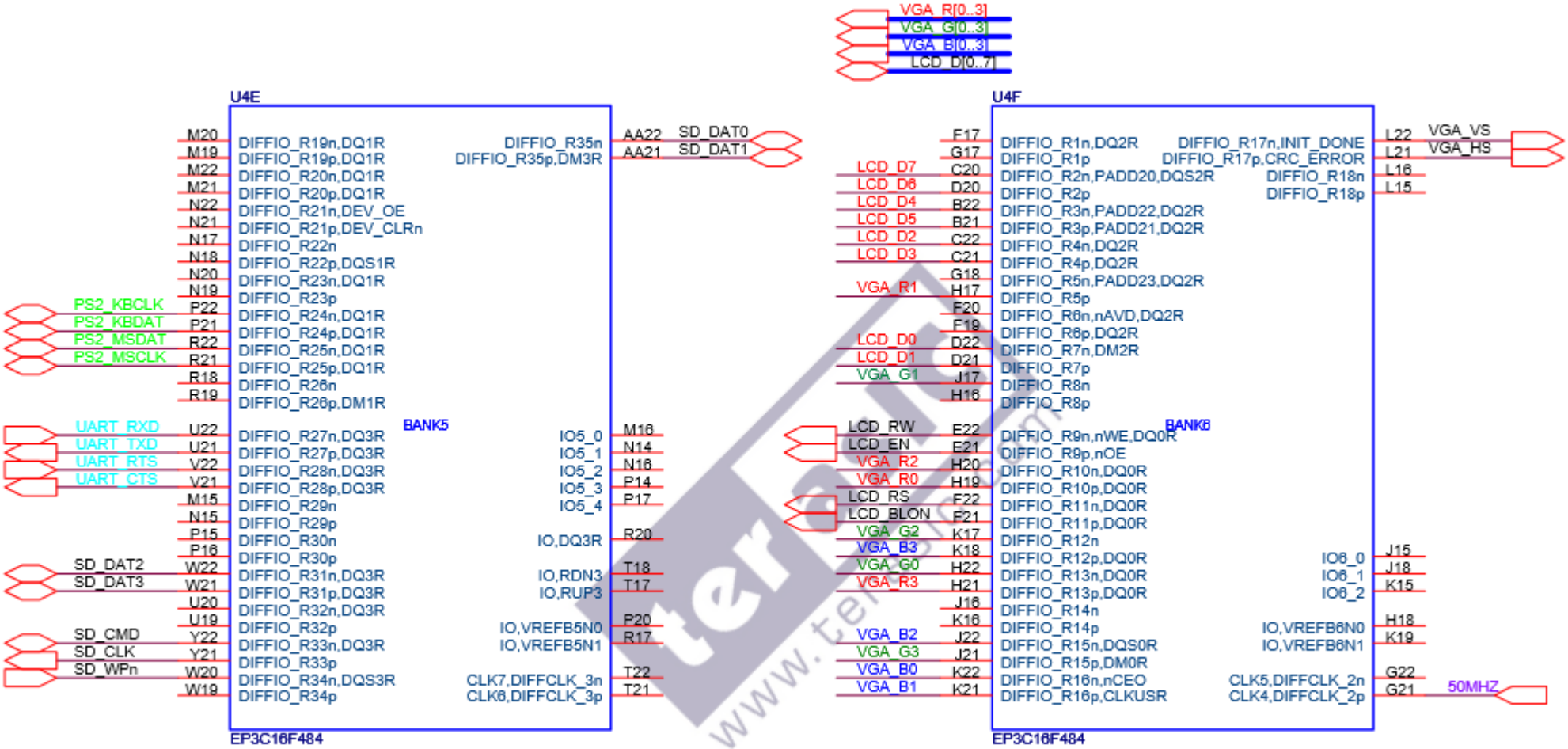


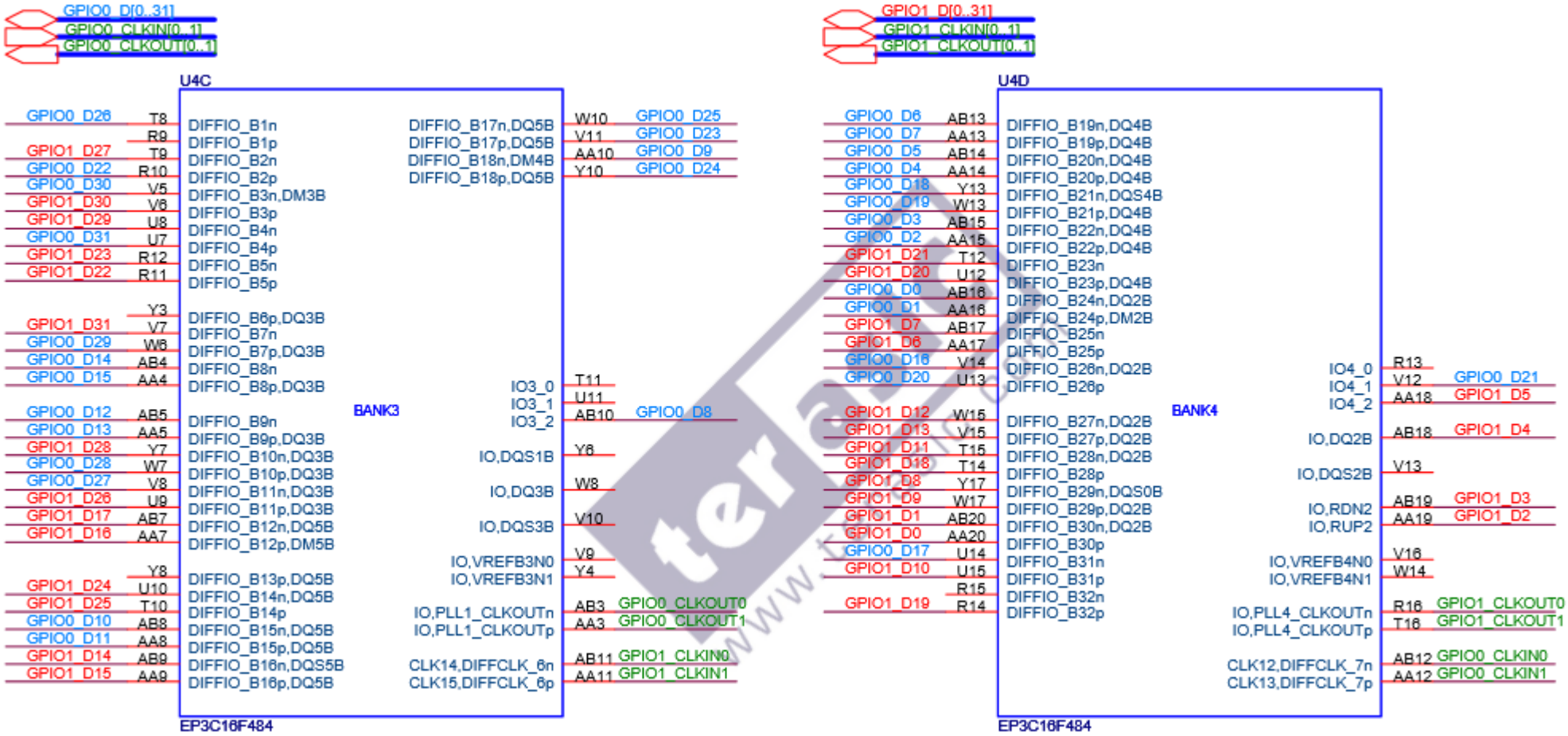


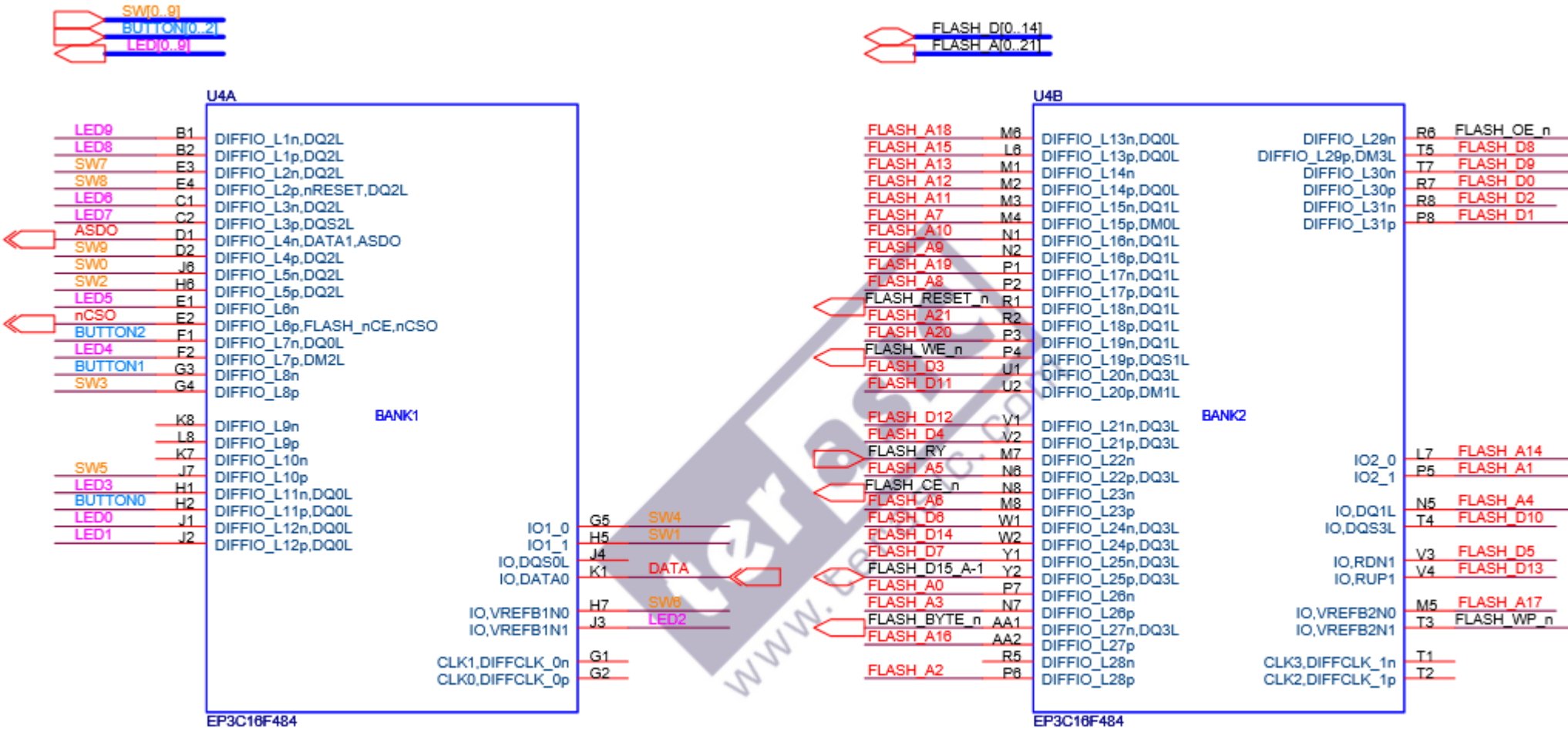

BUTTON[0..2]
SW[0..9]
VCC33
RN18
120
RN19
120
R26
R27
120
120
SW0
SW1
SW2
SW3
SW4
SW5
SW6
SW7
SW8
SW9
SLIDE SW


HEX0_DIO_6L
HEX1_DIO_6L
HEX2_DIO_6L
HEX3_DIO_6L
RN7
RN8
RN9
RN10
RN11
RN12
RN13
RN14
1K
HEX0
HEX1
HEX2
HEX3
7Segment Display
VCC33


HEX0_D[0..6]
HEX1_D[0..6]
HEX2_D[0..6]
HEX3_D[0..6]
U4G
HEX0_D0 E11
HEX0_D1 F11
HEX1_D0 A13
HEX1_D1 B13
HEX1_D3 A14
HEX1_D4 B14
HEX1_D2 C13
HEX0_DP D13
HEX1_D6 A15
HEX1_DP B15
HEX0_D4 G12
HEX0_D6 F13
HEX0_D3 H13
HEX0_D2 H12
HEX1_D5 E14
HEX0_D5 F12
HEX2_D1 A16
HEX2_D2 B16
HEX2_D4 A17
HEX2_D5 B17
HEX2_D0 D15
HEX2_D3 E15
HEX2_DP A18
HEX3_D0 B18
H15
H14
HEX3_D4 C19
HEX3_D5 D19
HEX3_D6 G15
HEX2_D6 F14
HEX3_D1 F15
HEX3_DP G16
DIFFIO_T16n,PADD13
DIFFIO_T16p,PADD14,DM4T
DIFFIO_T17n,PADD11,DQ4T
DIFFIO_T17p,PADD12,DQS4T
DIFFIO_T18n,PADD9,DQ4T
DIFFIO_T18p,PADD10,DQ4T
DIFFIO_T19n,PADD7
DIFFIO_T19p,PADD8,DQ4T
DIFFIO_T20n,PADD5,DQ4T
DIFFIO_T20p,PADD6,DQ4T
DIFFIO_T21n
DIFFIO_T21p,PADD4,DQS2T
DIFFIO_T22n
DIFFIO_T22p
DIFFIO_T23n,PADD3,DQ4T
DIFFIO_T23p
BANK7
DIFFIO_T24n,DM2T
DIFFIO_T24p,DQ4T
DIFFIO_T25n,PADD1,DQ2T
DIFFIO_T25p,PADD2
DIFFIO_T26n
DIFFIO_T26p,DQ2T
DIFFIO_T27n,DQ2T
DIFFIO_T27p,PADD0
DIFFIO_T28n
DIFFIO_T28p
DIFFIO_T29n,DQ2T
DIFFIO_T29p,DQ2T
DIFFIO_T30n
DIFFIO_T30p,DQS0T
DIFFIO_T31n,DQ2T
DIFFIO_T31p
DIFFIO_T32n F16
DIFFIO_T32p,DQ2T E16
IO7_0 E12
IO7_1 E13
IO7_2 G13
IO7_3 G14
IO,DQ2T C17
IO,RDN4 A19 HEX3_D2
IO,RUP4 B19 HEX3_D3
IO,VREFB7N0 D17
IO,VREFB7N1 C15
IO,PLL2_CLKOUTn A20
IO,PLL2_CLKOUTp B20
CLK8,DIFFCLK_5n A12
CLK9,DIFFCLK_5p B12 50MHZ
EP3C16F484
DRAM_D[0..15]
DRAM_A[0..12]
DRAM_DQM[0..1]
DRAM_BA[0..1]
U4H
DRAM_RAS_n F7
DRAM_CS_n G7
DRAM_A3 C3
DRAM_A0 C4
DRAM_A1 A3
DRAM_A2 B3
DRAM_D7 F8
DRAM_CAS_n G8
DRAM_BA1 A4
DRAM_A10 B4
DRAM_D4 F9
DRAM_D15 F10
DRAM_D2 H10
DRAM_D6 H9
DRAM_D1 G10
DRAM_D5 G9
DRAM_A12 C8
DRAM_A8 C7
DRAM_A7 A6
DRAM_A6 B6
DRAM_A11 A7
DRAM_A9 B7
DRAM_D8 A8
DRAM_DQM1 B8
DRAM_D10 A9
DRAM_D9 B9
DRAM_D13 A10
DRAM_D12 B10
DRAM_D0 D10
DRAM_D14 E10
DIFFIO_T1n,DM3T
DIFFIO_T1p
DIFFIO_T2n,DQ3T
DIFFIO_T2p,DATA12,DQS1T
DIFFIO_T3n,DATA10,DQ3T
DIFFIO_T3p,DATA11,DQ3T
DIFFIO_T4n,DATA9,DQ3T
DIFFIO_T4p
DIFFIO_T5n,DQ3T
DIFFIO_T5p,DATA8,DQ3T
DIFFIO_T6n
DIFFIO_T6p,DATA6,DQ3T
DIFFIO_T7n
DIFFIO_T7p
DIFFIO_T8n
DIFFIO_T8p
BANK8
DIFFIO_T9n,DATA14,DQS3T
DIFFIO_T9p,DATA13,DM5T
DIFFIO_T10n,PADD19,DQ5T
DIFFIO_T10p,DATA15,DQ5T
DIFFIO_T11n,PADD18,DQ5T
DIFFIO_T11p,DATA4,DQ5T
DIFFIO_T12n,DATA2,DQ5T
DIFFIO_T12p,DATA3,DQ5T
DIFFIO_T13n,PADD16,DQ5T
DIFFIO_T13p,PADD17,DQS5T
DIFFIO_T14n,DQ5T
DIFFIO_T14p,PADD15
DIFFIO_T15n,DQ5T
DIFFIO_T15p
IO8_0 B5 DRAM_BA0
IO8_1 C10 DRAM_D11
IO8_2 E7 DRAM_DQM0
IO8_3 G11
IO8_4 H11
IO,DATA5,DQ3T A5 DRAM_A4
IO,DATA7,DQ3T C6 DRAM_A5
IO,VREFB8N0 E9 DRAM_D3
IO,VREFB8N1 D6 DRAM_WE_n
IO,PLL3_CLKOUTn E6 DRAM_CKE
IO,PLL3_CLKOUTp E5 DRAM_CLK
CLK10,DIFFCLK_4n A11
CLK11,DIFFCLK_4p B11
EP3C16F484




GPIO0_D[0..31]
GPIO0_CLKIN[0..1]
GPIO0_CLKOUT[0..1]
U4C
GPIO0_D26 T8 DIFFIO_B1n
R9 DIFFIO_B1p
GPIO1_D27 T9 DIFFIO_B2n
GPIO0_D22 R10 DIFFIO_B2p
GPIO0_D30 V5 DIFFIO_B3n,DM3B
GPIO1_D30 V6 DIFFIO_B3p
GPIO1_D29 U8 DIFFIO_B4n
GPIO0_D31 U7 DIFFIO_B4p
GPIO1_D23 R12 DIFFIO_B5n
GPIO1_D22 R11 DIFFIO_B5p
Y3 DIFFIO_B6p,DQ3B
GPIO1_D31 V7 DIFFIO_B7n
GPIO0_D29 W6 DIFFIO_B7p,DQ3B
GPIO0_D14 AB4 DIFFIO_B8n
GPIO0_D15 AA4 DIFFIO_B8p,DQ3B
GPIO0_D12 AB5 DIFFIO_B9n
GPIO0_D13 AA5 DIFFIO_B9p,DQ3B
GPIO1_D28 Y7 DIFFIO_B10n,DQ3B
GPIO0_D28 W7 DIFFIO_B10p,DQ3B
GPIO0_D27 V8 DIFFIO_B11n,DQ3B
GPIO1_D26 U9 DIFFIO_B11p,DQ3B
GPIO1_D17 AB7 DIFFIO_B12n,DQ5B
GPIO1_D16 AA7 DIFFIO_B12p,DM5B
Y8 DIFFIO_B13p,DQ5B
GPIO1_D24 U10 DIFFIO_B14n,DQ5B
GPIO1_D25 T10 DIFFIO_B14p
GPIO0_D10 AB8 DIFFIO_B15n,DQ5B
GPIO0_D11 AA8 DIFFIO_B15p,DQ5B
GPIO1_D14 AB9 DIFFIO_B16n,DQS5B
GPIO1_D15 AA9 DIFFIO_B16p,DQ5B
BANK3
DIFFIO_B17n,DQ5B W10 GPIO0_D25
DIFFIO_B17p,DQ5B V11 GPIO0_D23
DIFFIO_B18n,DM4B AA10 GPIO0_D9
DIFFIO_B18p,DQ5B Y10 GPIO0_D24
IO3_0 T11
IO3_1 U11
IO3_2 AB10 GPIO0_D8
IO,DQS1B Y6
IO,DQ3B W8
IO,DQS3B V10
IO,VREFB3N0 V9
IO,VREFB3N1 Y4
IO,PLL1_CLKOUTn AB3 GPIO0_CLKOUT0
IO,PLL1_CLKOUTp AA3 GPIO0_CLKOUT1
CLK14,DIFFCLK_6n AB11 GPIO1_CLKIN0
CLK15,DIFFCLK_6p AA11 GPIO1_CLKIN1
EP3C16F484
GPIO1_D[0..31]
GPIO1_CLKIN[0..1]
GPIO1_CLKOUT[0..1]
U4D
GPIO0_D6 AB13 DIFFIO_B19n,DQ4B
GPIO0_D7 AA13 DIFFIO_B19p,DQ4B
GPIO0_D5 AB14 DIFFIO_B20n,DQ4B
GPIO0_D4 AA14 DIFFIO_B20p,DQ4B
GPIO0_D18 Y13 DIFFIO_B21n,DQS4B
GPIO0_D19 W13 DIFFIO_B21p,DQ4B
GPIO0_D3 AB15 DIFFIO_B22n,DQ4B
GPIO0_D2 AA15 DIFFIO_B22p,DQ4B
GPIO1_D21 T12 DIFFIO_B23n
GPIO1_D20 U12 DIFFIO_B23p,DQ4B
GPIO0_D0 AB16 DIFFIO_B24n,DQ2B
GPIO0_D1 AA16 DIFFIO_B24p,DM2B
GPIO1_D7 AB17 DIFFIO_B25n
GPIO1_D6 AA17 DIFFIO_B25p
GPIO0_D16 V14 DIFFIO_B26n,DQ2B
GPIO0_D20 U13 DIFFIO_B26p
GPIO1_D12 W15 DIFFIO_B27n,DQ2B
GPIO1_D13 V15 DIFFIO_B27p,DQ2B
GPIO1_D11 T15 DIFFIO_B28n,DQ2B
GPIO1_D18 T14 DIFFIO_B28p
GPIO1_D8 Y17 DIFFIO_B29n,DQS0B
GPIO1_D9 W17 DIFFIO_B29p,DQ2B
GPIO1_D1 AB20 DIFFIO_B30n,DQ2B
GPIO1_D0 AA20 DIFFIO_B30p
GPIO0_D17 U14 DIFFIO_B31n
GPIO1_D10 U15 DIFFIO_B31p
R15 DIFFIO_B32n
GPIO1_D19 R14 DIFFIO_B32p
BANK4
IO4_0 R13
IO4_1 V12 GPIO0_D21
IO4_2 AA18 GPIO1_D5
IO,DQ2B AB18 GPIO1_D4
IO,DQS2B V13
IO,RDN2 AB19 GPIO1_D3
IO,RUP2 AA19 GPIO1_D2
IO,VREFB4N0 V16
IO,VREFB4N1 W14
IO,PLL4_CLKOUTn R16 GPIO1_CLKOUT0
IO,PLL4_CLKOUTp T16 GPIO1_CLKOUT1
CLK12,DIFFCLK_7n AB12 GPIO0_CLKIN0
CLK13,DIFFCLK_7p AA12 GPIO0_CLKIN1
EP3C16F484


SW[0..9]
BUTTON[0..2]
LED[0..9]
U4A
LED9 B1 DIFFIO_L1n,DQ2L
LED8 B2 DIFFIO_L1p,DQ2L
SW7 E3 DIFFIO_L2n,DQ2L
SW8 E4 DIFFIO_L2p,nRESET,DQ2L
LED6 C1 DIFFIO_L3n,DQ2L
LED7 C2 DIFFIO_L3p,DQS2L
ASDO D1 DIFFIO_L4n,DATA1,ASDO
SW9 D2 DIFFIO_L4p,DQ2L
SW0 J6 DIFFIO_L5n,DQ2L
SW2 H6 DIFFIO_L5p,DQ2L
LED5 E1 DIFFIO_L6n
nCSO E2 DIFFIO_L6p,FLASH_nCE,nCSO
BUTTON2 F1 DIFFIO_L7n,DQ0L
LED4 F2 DIFFIO_L7p,DM2L
BUTTON1 G3 DIFFIO_L8n
SW3 G4 DIFFIO_L8p
BANK1
K8 DIFFIO_L9n
L8 DIFFIO_L9p
K7 DIFFIO_L10n
SW5 J7 DIFFIO_L10p
LED3 H1 DIFFIO_L11n,DQ0L
BUTTON0 H2 DIFFIO_L11p,DQ0L
LED0 J1 DIFFIO_L12n,DQ0L
LED1 J2 DIFFIO_L12p,DQ0L
IO1_0 G5 SW4
IO1_1 H5 SW1
IO,DQS0L J4
IO,DATA0 K1 DATA
IO,VREFB1N0 H7 SW6
IO,VREFB1N1 J3 LED2
CLK1,DIFFCLK_0n G1
CLK0,DIFFCLK_0p G2
EP3C16F484
FLASH_D[0..14]
FLASH_A[0..21]
U4B
FLASH_A18 M6 DIFFIO_L13n,DQ0L
FLASH_A15 L6 DIFFIO_L13p,DQ0L
FLASH_A13 M1 DIFFIO_L14n
FLASH_A12 M2 DIFFIO_L14p,DQ0L
FLASH_A11 M3 DIFFIO_L15n,DQ1L
FLASH_A7 M4 DIFFIO_L15p,DM0L
FLASH_A10 N1 DIFFIO_L16n,DQ1L
FLASH_A9 N2 DIFFIO_L16p,DQ1L
FLASH_A19 P1 DIFFIO_L17n,DQ1L
FLASH_A8 P2 DIFFIO_L17p,DQ1L
FLASH_RESET_n R1 DIFFIO_L18n,DQ1L
FLASH_A21 R2 DIFFIO_L18p,DQ1L
FLASH_A20 P3 DIFFIO_L19n,DQ1L
FLASH_WE_n P4 DIFFIO_L19p,DQS1L
FLASH_D3 U1 DIFFIO_L20n,DQ3L
FLASH_D11 U2 DIFFIO_L20p,DM1L
BANK2
FLASH_D12 V1 DIFFIO_L21n,DQ3L
FLASH_D4 V2 DIFFIO_L21p,DQ3L
FLASH_RY M7 DIFFIO_L22n
FLASH_A5 N6 DIFFIO_L22p,DQ3L
FLASH_CE_n N8 DIFFIO_L23n
FLASH_A6 M8 DIFFIO_L23p
FLASH_D6 W1 DIFFIO_L24n,DQ3L
FLASH_D14 W2 DIFFIO_L24p,DQ3L
FLASH_D7 Y1 DIFFIO_L25n,DQ3L
FLASH_D15_A-1 Y2 DIFFIO_L25p,DQ3L
FLASH_A0 P7 DIFFIO_L26n
FLASH_A3 N7 DIFFIO_L26p
FLASH_BYTE_n AA1 DIFFIO_L27n,DQ3L
FLASH_A16 AA2 DIFFIO_L27p
R5 DIFFIO_L28n
FLASH_A2 P6 DIFFIO_L28p
DIFFIO_L29n R6 FLASH_OE_n
DIFFIO_L29p,DM3L T5 FLASH_D8
DIFFIO_L30n T7 FLASH_D9
DIFFIO_L30p R7 FLASH_D0
DIFFIO_L31n R8 FLASH_D2
DIFFIO_L31p P8 FLASH_D1
IO2_0 L7 FLASH_A14
IO2_1 P5 FLASH_A1
IO,DQ1L N5 FLASH_A4
IO,DQS3L T4 FLASH_D10
IO,RDN1 V3 FLASH_D5
IO,RUP1 V4 FLASH_D13
IO,VREFB2N0 M5 FLASH_A17
IO,VREFB2N1 T3 FLASH_WP_n
CLK3,DIFFCLK_1n T1
CLK2,DIFFCLK_1p T2
EP3C16F484