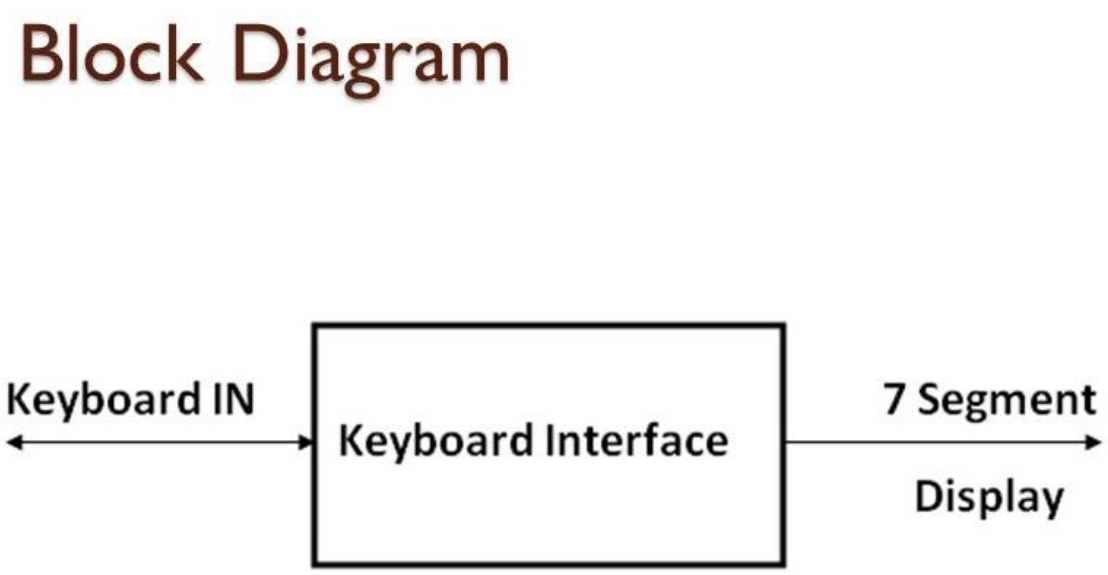
پروژه آزمایشگاه معماری

محمد صالح پژند 400521171 آرمین افضلی 400521054

هدف پروژه:

ارسال رقم توسط کیبورد به برد و نمایش آن بر روی seven segment.

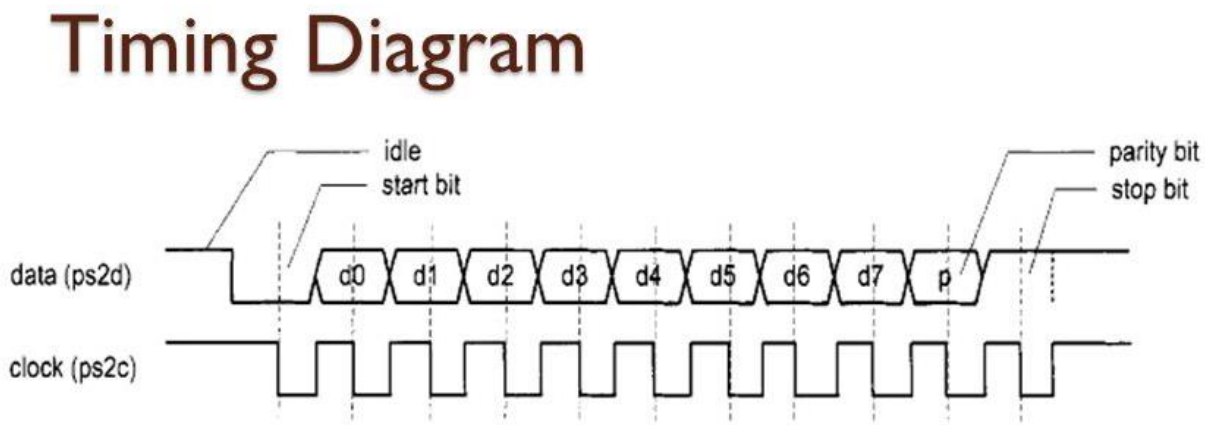


شرح پروژه:

مراحل انجام این پروژه مانند ترکیبی از آزمایشات گذشته است با یک سری تفاوت های جزعی:

- خواندن بیت به صورت سریالی است، مانند خواند بیت از rx، با این تفاوت که در اینجا از پورت ps2 علاوه بر داده، clock کیبورد نیز گرفته می شود (بر خلاف rx که asynchronous است، کیبورد به صورت synchronous عمل می کند)؛ به این صورت عمل می کند که ps2_clock inout است و ما در ابتدا یک بار مقدار ps2_clock را یک کرده و از آن به بعد، به عنوان clock برای process اصلی مورد استفاده قرار می گیرد.

- وجود parity bit نیز از تفاوت های دیگر این process است، parity bit کیبورد از نوع odd parity است، به این معنی که تعداد یک های دیتای ورودی باید فرد باشد و در غیر این صورت داده ورودی صحت ندارد.



- همانطور که مشاهده می شود، ps2 با لبه پایین رونده clock کار می کند و دارای دو بیت start و stop، 8 بیت داده و یک بیت parity است.

Scan Codes:

که برای تعیین محتوای داده ورودی استفاده شده است.

نمایش بر روی seven segment هم که در آزمایشات گذشته توضیح داده شده است، فقط در این آزمایش تنها به یک seven segment نیاز داریم.

Code:

تعیین ورودی و خروجی ها و همچنین سیگنال های مورد استفاده:

```
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
USE IEEE.STD_LOGIC_ARITH.ALL;
USE IEEE.STD_LOGIC_UNSIGNED.ALL;

ENTITY PS2 IS
    PORT (
        CLK : IN STD_LOGIC;
        DATA : IN STD_LOGIC;
        PS2_Clock : INOUT STD_LOGIC;
        sel_seg : OUT STD_LOGIC;
        output : OUT STD_LOGIC_VECTOR (6 DOWNTO 0));
END PS2;

ARCHITECTURE Behavioral OF PS2 IS

    SIGNAL clock_10ms : STD_LOGIC := '0';
    SIGNAL PS2_DATA : STD_LOGIC_VECTOR (7 DOWNTO 0) := (OTHERS => '0');
```

CLK:

همان gclock برد است.

DATA:

بیت دریافتی از کیبورد.

PS2_Clock:

Clock کیبورد که برای synchronization استفاده می شود.

Sel_seg:

تعیین کننده seven segment مورد استفاده.

Output:

مقدار داده شده به seven segment برای نمایش رقم، متناسب با داده ورودی.

Clock_10ms:

برای seven segment refresh استفاده می شود.

PS2_DATA:

8 بیت دریافتی.

تعیین مقدار خروجی با استفاده از scan code ها:

```
WITH PS2_DATA SELECT
    output <= "0111111" WHEN x"45",
    "0000110" WHEN x"16",
    "1011011" WHEN x"1e",
    "1001111" WHEN x"26",
    "1100110" WHEN x"25",
    "1101101" WHEN x"2e",
    "1111101" WHEN x"36",
    "0000111" WHEN x"3d",
    "1111111" WHEN x"3e",
    "1101111" WHEN x"46",
    "0000000" WHEN OTHERS;
```

Clock 10ms process:

```
PROCESS (CLK)
    VARIABLE count_div : INTEGER RANGE 0 TO 50000 := 0;
BEGIN
    IF (rising_edge(CLK)) THEN
        IF (count_div < 50000) THEN
            count_div := count_div + 1;
        ELSE
            count_div := 0;
            clock10ms <= NOT clock10ms;
        END IF;
    END IF;
END PROCESS;
```

Seven segment refresh process:

```
PROCESS (clock_10ms)
BEGIN
    IF rising_edge(clock_10ms) THEN
        IF (sel_seg = '0') THEN
            sel_seg <= '1';
        ELSE
            sel_seg <= '0';
        END IF;
    END IF;
END PROCESS;
```

Setting ps2 clock process:

```
PROCESS (CLK)
    VARIABLE counter : INTEGER RANGE 0 TO 100000 := 0;
    VARIABLE is_clock_set : STD_LOGIC := '0';
BEGIN
    IF (rising_edge(CLK) AND is_clock_set = '0') THEN
        IF (counter < 750000) THEN
            counter := counter + 1;
        ELSE
            counter := 0;
            PS2_Clock <= '1';
            is_clock_set = '1';
        END IF;
    END IF;
END PROCESS;
```

همانطور که مشاهده می شود، پس از یک بار set کردن clock دیگر به آن خروجی داده نمی شود.

Main process:

```
PROCESS (PS2_Clock)
    VARIABLE PS2_DATA_TEMP : STD_LOGIC_VECTOR(7 DOWNTO 0) := (OTHERS => '0');
    VARIABLE counter : INTEGER RANGE 0 TO 10 := 0;
    VARIABLE finish : STD_LOGIC := '1';
    VARIABLE one_c : INTEGER RANGE 0 TO 10 := 0;
BEGIN
    IF falling_edge(PS2_Clock) THEN
        IF finish = '0' THEN
            IF counter < 8 THEN
                PS2_DATA_TEMP(counter) := DATA;

                IF (DATA = '1') THEN
                    one_c := one_c + 1;
                END IF;

                counter := counter + 1;
            ELSIF counter = 8 THEN
                IF DATA = '1' THEN
                    one_c := one_c + 1;
                END IF;

                IF one_c REM 2 = '1' THEN
                    PS2_DATA <= PS2_DATA_TEMP;
                END IF;

                counter := counter + 1;
            ELSIF counter = 9 THEN
                counter := 0;
                finish := '1';
                one_c := 0;
            END IF;
        ELSIF DATA = '0' THEN
            finish = '0';
        END IF;
    END IF;
END PROCESS;
```

تا زمانی که بیت صفر که همان بیت start است، دریافت نشده است process خواندن دیتا آغاز نمی شود.

8 بیت ابتدایی مقدار دیتای ورودی را مشخص می کنند که توسط one_c مقدار یک های آن شمرده می شود و بیت 9 که همان parity است، نیز در صورت یک بودن شمرده می شود، فرد بودن یک های کلی به معنای valid بودن دیتای ورودی است، پس مراحل parse شدن بر روی آن صورت می گیرد.

زمانی هم که counter 9 می شود، به معنای اتمام این مرحله از دیتا است و منتظر دریافت start bit بعدی می شود.

Constraints:

```
NET "CLK" LOC = P184;
NET "CLK" CLOCK_DEDICATED_ROUTE = FALSE;
NET "PS2_Clock" LOC = P92;
NET "DATA" LOC = P83;
NET "output[0]" LOC = P10;
NET "output[1]" LOC = P7;
NET "output[2]" LOC = P11;
NET "output[3]" LOC = P5;
NET "output[4]" LOC = P4;
NET "output[5]" LOC = P12;
NET "output[6]" LOC = P9;
NET "selseg" LOC = P15;
```

منابع استفاده شده:

https://slideplayer.com/slide/5949770/