Report

Realization of a virtual Private Branch Exchange(PBX) on A MAX II kit
using VHDL coding.

By Ujjval Shah(05BEC080)
Priya Vaya(05BEC090)

A **Private Branch eXchange** (PBX) is a telephone exchange that serves a particular business or office, as opposed to one that a common carrier or telephone company operates for many businesses or for the general public. PBXs make connections among the internal telephones of a private organization , usually a business , and also connect them to the public switched telephone network (PSTN) via trunk lines.

The entire process occurring at the PBX can be divided into multiple modules. The realization has been carried for a total of 5 users .

Module 1: USERS' HOOK SCANNING
The pbx constantly scans the user's lines for detecting an off hook condition(i.e. the receiver has been lifted).If the i^{th} user is off hook, the flag pertaining to that user(Busy flag) is set, indicating its off hook status. This is followed by the reception of a dial tone by that user. Also an flag indicating its 'engaged' status is set, so that any other user can't get access to that particular user. The flag indicating ringing tone is reset for obvious reasons. When its in the on hook condition, the user busy flag, Engaged tone flag, dial tone flag are all reset .Once the user receives the dial tone, he can dial a number in the keypad. Hence the digit dialing enable flag is set.

Module 2: LEDS
Post the above process, if the dial tone flag for that user is set, then the LED pertaining to that user indicates it is receiving the dial pulse. Subsequently, the engaged status and the ringing tone status is reflected by the LEDS.

Module 3: Dialing the number(Keypad Scanning)
Particular numbers have been assigned to the various users. The 1^{st} user's number is 00001,the 2^{nd} user's number is 00010 and so on. If the digit dialing is enabled, the number is dialed through the keypad. Suppose, the number 00001 is dialed by a particular user, then the scanner will check for its busy flag's status. If it is in reset state, then the i^{th} user and the a^{th} user receive a ringing tone, setting the dial tone offhand the connection is made ,after which the digit dialing flag is set off. The same process is repeated if the 2^{nd},3^{rd} etc user's number is dialed.
However, if the ith user is trying to reach the ath user which has its busy flag set, then the engaged tone is received by that user. Only after on hooking the receiver can the user dial a new number. Hence the digit enable flag is reset momentarily and then set once it enters the loop defined by the module 1.

For this purpose, we require four signals viz. ringing tone, dial tone, clock and the engaged tone. The clock is provided internally from the MAX II hardware. The rest of them are supplied by microcontroller ATMEGA16.The VHDL coding is as follows.

```
1   -- Hardware/Software implementation of a Public Branch Exchange --
```

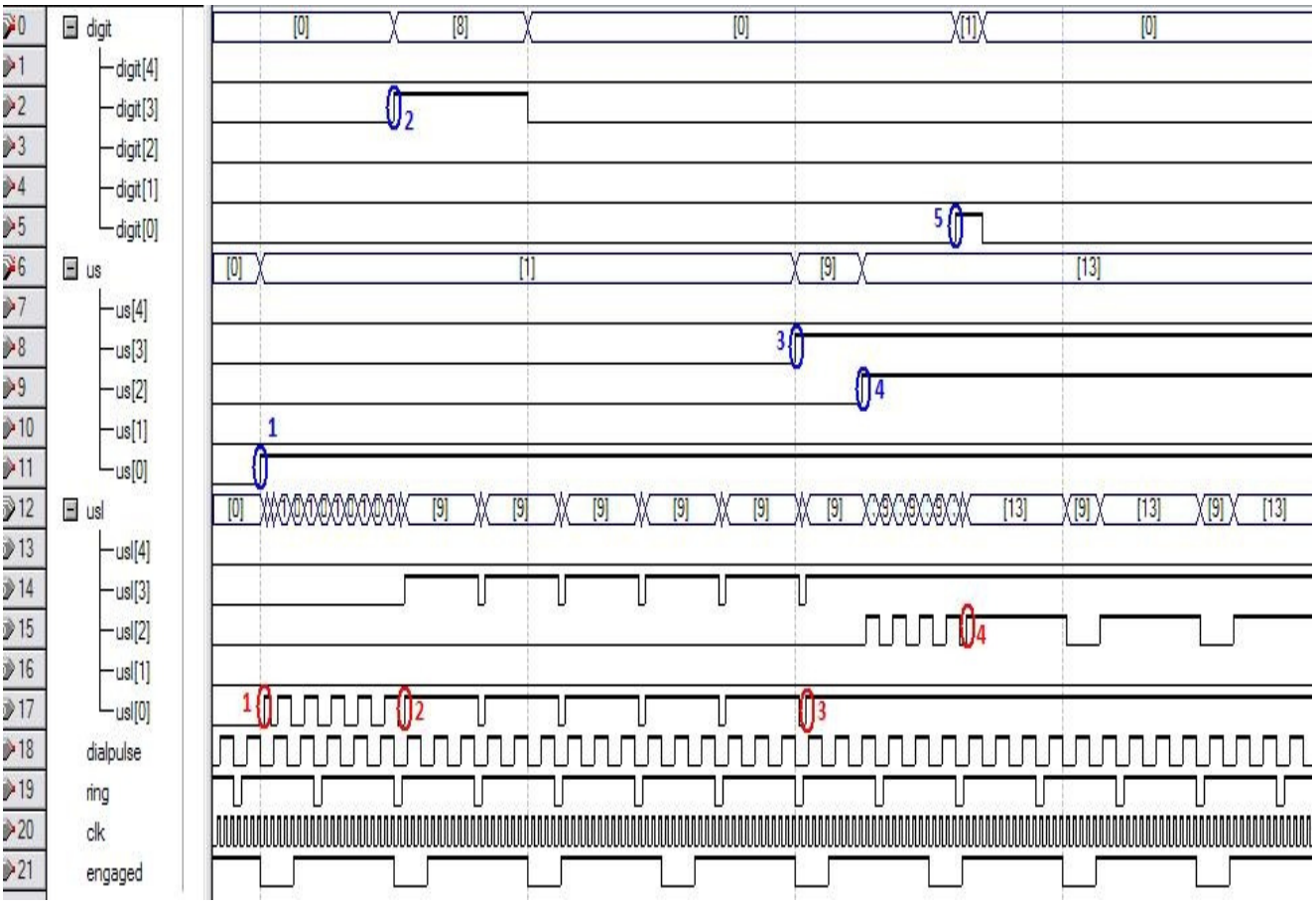
```vhdl
-- on an MAX II: EPM240T100C5 kit using VHDL coding --
-- 05 BEC 080 --
-- 05 BEC 093 --
library ieee;
use ieee.std_logic_1164.all;

entity pbx2 is
port(clk, ring, engaged, dialpulse :in bit;
     us, digit : in bit_vector(4 downto 0);
     usl : out bit_vector(4 downto 0):="00000");
end pbx2;

 -- us -> user
 -- usl -> user led
 -- digit -> Digit entered from the keypad
 -- ring, engaged, dialpulse -> respective signals on the leds

architecture sw of pbx2 is
  signal B, usld, usle, uslr: bit_vector(4 downto 0):="00000";
  signal toggle : bit_vector(4 downto 0):="11111";
  signal digiten : bit;

  -- usld -> Dial Tone Flag for that user
  -- uslr -> Ringing Tone Flag for that user
  -- usle -> Engaged Tone Flag for that user
  -- B -> Busy Flag for that user
  -- digiten -> Keypad enable Flag
  -- toggle -> Toggle Flag to detect the change in the on/off hook
condition

  begin
    process (clk)
      variable a : integer:=0;
      variable call : bit_vector(4 downto 0):="00000";
  begin
  if clk'event and clk='1'then
  --============================
  -- Users' Hook Scanning --
  --============================
  Scanning: for i in 0 to 4 loop

  if us(i)='0' then -- on hook condition
     if toggle(i)='1' then
          usl(i)<='0';
          usld(i)<='0';
          usle(i)<='0';
          uslr(i)<='0';
          call(i):='0';
          B(i)<='0';
     end if;
  toggle(i)<='0';
  ----------------------------------------
  elsif us(i)='1' then
       usl(i)<='1'; -- off hook condition
  if toggle(i)='0' then
       usle(i)<='0';
       uslr(i)<='0';
       uslr(a)<='0';
  if call(i)='1' then -- receiving a call
       usl(i)<='1';
       usl(a)<='1';
       B(i)<='1';
 elsif call(i)='0' then -- making a call
       B(i)<='1';
```

```
65          a:=i;
66          digiten<='1';
67          usld(i)<='1';
68    end if;
69          toggle(i)<='1';
70    end if;
71    end if;
72
73   end loop Scanning;
74  --============================
75  -- Users' Hook Scanning Ends --
76  --============================
77   LEDS: for j in 0 to 4 loop
78
79    if usld(j)='1' then
80         usl(j)<=dialpulse;
81    end if;
82
83    if uslr(j)='1' then
84         usl(j)<=ring;
85    end if;
86
87    if usle(j)='1' then
88         usl(j)<=engaged;
89    end if;
90
91   end loop LEDS;
92  --============================
93  -- Keypad Scanning --
94  --============================
95   if digiten='1' then
96    case digit is
97       when "00001" => --#1
98           if B(0)='0' then
99               uslr(0)<='1';
100              uslr(a)<='1';
101              usld(a)<='0';
102              call(0):='1';
103              digiten<='0';
104          elsif B(0)='1' then
105              usle(a)<='1';
106              digiten<='0';
107          end if;
108      when "00010" => --#2
109       if B(1)='0' then
110          uslr(1)<='1';
111          uslr(a)<='1';
112          usld(a)<='0';
113          call(1):='1';
114          digiten<='0';
115      elsif B(1)='1' then
116          usle(a)<='1';
117          digiten<='0';
118       end if;
119    when "00100" => --#3
120       if B(2)='0' then
121          uslr(2)<='1';
122          uslr(a)<='1';
123          usld(a)<='0';
124          call(2):='1';
125          digiten<='0';
126       elsif B(2)='1' then
127          usle(a)<='1';
128          digiten<='0';
```

```
129       end if;
130    when "01000" => --#4
131       if B(3)='0' then
132          uslr(3)<='1';
133          uslr(a)<='1';
134          usld(a)<='0';
135          call(3):='1';
136          digiten<='0';
137       elsif B(3)='1' then
138          usle(a)<='1';
139          digiten<='0';
140      end if;
141   when "10000" => --#5
142      if B(4)='0' then
143          uslr(4)<='1';
144          uslr(a)<='1';
145          usld(a)<='0';
146          call(4):='1';
147          digiten<='0';
148      elsif B(4)='1' then
149          usle(a)<='1';
150          digiten<='0';
151      end if;
152    when others =>
153         null;
154  end case;
155 end if;
156 --============================
157 -- Keypad Scanning Ends --
158 --============================
159 end if;
160 end process;
161 end sw;
```

Simulatio results:

1 (blue) - user 0 picks up
1 (r) - gets the dial pulse
2 (b) - dials #3
1 and 3 gets the ringing
3 picks up
both are connected
2 picks up and gets the dial
dials to 0 and gets the engaged tone


digit
digit[4]
digit[3]
digit[2]
digit[1]
digit[0]
us
us[4]
us[3]
us[2]
us[1]
us[0]
usl
usl[4]
usl[3]
usl[2]
usl[1]
usl[0]
dialpulse
ring
clk
engaged
[0]
[8]
[0]
[1]
[0]
[0]
[1]
[9]
[13]
[0]
[9]
[13]
[9]
[13]
[9]
[13]
1
2
3
4
5