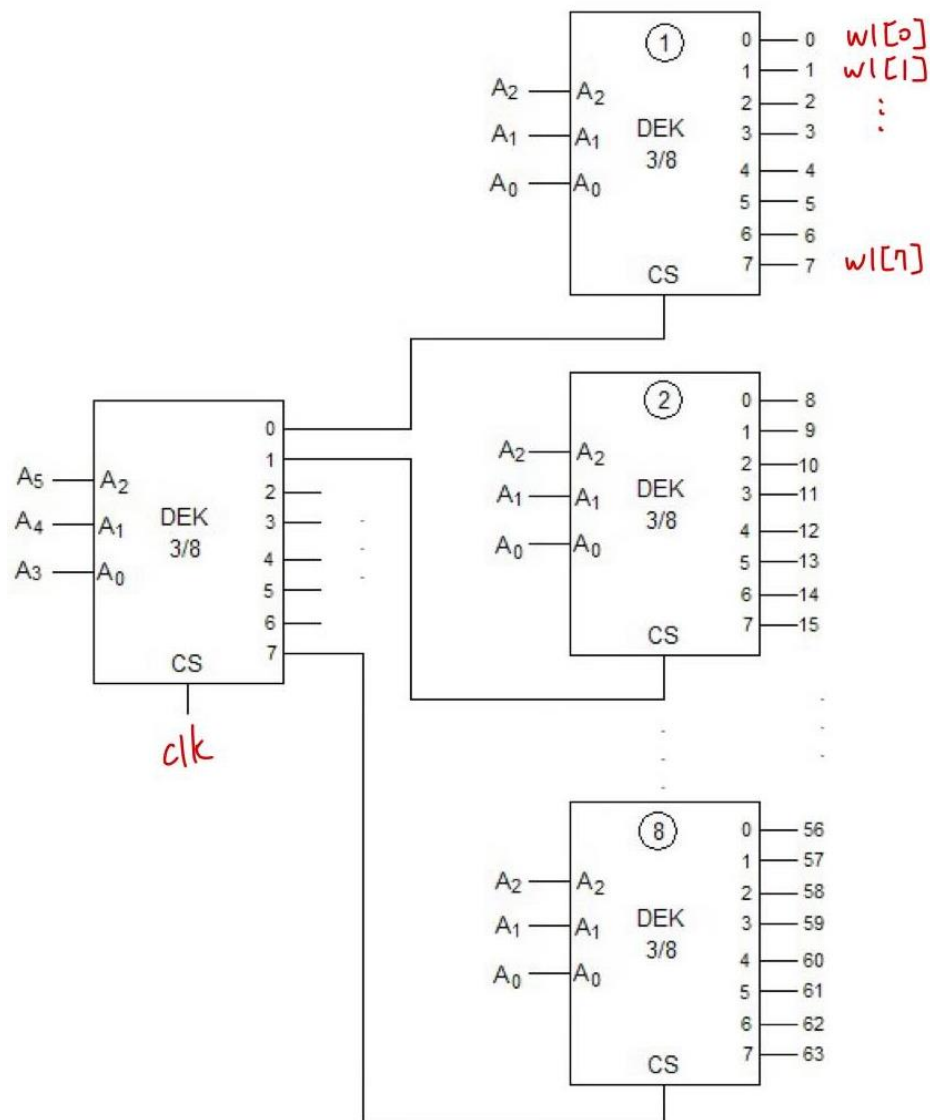


Architecture :



使用 9 個 3to8_decoder 來設計 6to64_decoder，將其中一個的輸出接上剩下 8 個的 enable，如此便能控制 WL 輸出，再將 clk 接上 MSB 端 decoder 的 enable 來達到 decoder 功用。

Python :

```

1  import math
2
3  num_addr = 4
4
5  def vector():
6      vf = open("input.vec","w")
7      vf.write("RADIX 1 1 1 1 1 1\n")
8      vf.write("VNAME clk Addr[5] Addr[4] Addr[3] Addr[2] Addr[1] Addr[0]\n")
9      vf.write("IO I I I I I I\n")
10     vf.write("tunit ns\n")
11     vf.write("slope 0.05\n")
12     vf.write("tdelay 0.0\n")
13     vf.write("vih 0.7\n")
14     vf.write("vil 0\n")
15     count = 0
16     for i in range(64):
17         vf.write(str(count*1) + "\t\t" + "0 " + ("0 " if (int)(i/32)%2 == 0 else "1 ")
18                 + ("0 " if (int)(i/16)%2 == 0 else "1 ") + ("0 " if (int)(i/8)%2 == 0 else "1 ")
19                 + ("0 " if (int)(i/4)%2 == 0 else "1 ") + ("0 " if (int)(i/2)%2 == 0 else "1 ")
20                 + ("0 " if i%2 == 0 else "1 ") + "\n")
21         vf.write(str(count*1+0.5) + "\t\t" + "1 " + ("0 " if (int)(i/32)%2 == 0 else "1 ")
22                 + ("0 " if (int)(i/16)%2 == 0 else "1 ") + ("0 " if (int)(i/8)%2 == 0 else "1 ")
23                 + ("0 " if (int)(i/4)%2 == 0 else "1 ") + ("0 " if (int)(i/2)%2 == 0 else "1 ") + ("0 " if i%2 == 0 else "1 ") + "\n")
24         count+=1
25
26     vf.close()

```

```

29  #####
30  #-----CHOS-----#
31  #####
32  def INV(file):
33      string = ".subckt inverter vin vout\n"
34      string = string + "M1 vout vin GND x nmos_sram\tm=1\n"
35      string = string + "M2 vout vin VDD x pmos_sram\tm=1\n"
36      string = string + ".ends\n"
37      string = string + ".subckt inv_chain vin vout\n"
38      string = string + "X1 vin net0 inverter\n"
39      string = string + "X2 net0 vout inverter\n"
40      string = string + ".ends\n"
41      file.write( string )
42
43      number_mos = 3
44      number_node = 0
45
46  def NAND(file,num_addr):
47      global number_mos,number_node
48      string = ".subckt nand_"+str(num_addr)
49      str1 = ""
50      for i in range(num_addr):
51          str1 = str1 + "\tin" + str(i) + "\t"
52      string = string + str1 + "\tout\n"
53
54      str1 = ""
55      for i in range(num_addr):
56          if i==0:
57              str1 = str1 + "M"+str(number_mos)+"\t" + "n" + str(number_node)+"\tin"+str(i) + "\tGND" + "\tx\tmos_sram\tm=1\n"
58              number_node = number_node + 1
59          elif i==(num_addr-1):
60              str1 = str1 + "M"+str(number_mos)+"\t" + "out"+"\tin"+str(i) + "\t" + "n" + str(i-1) + "\tx\tmos_sram\tm=1\n"
61          else:
62              str1 = str1 + "M"+str(number_mos)+"\t" + "n" + str(number_node) + "\tin"+str(i) + "\t" + "n" + str(number_node-1) + "\tx\tmos_sram\tm=1\n"
63              number_node = number_node + 1
64              number_mos = number_mos + 1
65
66      string = string + str1
67      str1 = ""
68      for i in range(num_addr):
69          str1 = str1 + "M"+str(number_mos)+"\tout"+"\tin"+str(i)+"\tVDD" + "\tx\tmos_sram\tm=1\n"
70          number_mos = number_mos + 1
71      string = string + str1 + ".ends\n"
72      file.write(string)

```

```

75  def AND(file,num_addr):
76      global number_node
77      string = ".subckt and_"+str(num_addr)
78      str1 = ""
79      for i in range(num_addr):
80          str1 = str1 + "\tin" + str(i) + "\t"
81      string = string + str1 + "\tout\n"
82
83      string = string + "x3" + str1 + "n" + str(number_node) + "\tnand_"+str(num_addr)+"\n"
84      string = string + "x4\t" + "n" + str(number_node) + "\tout\tinverter\n"
85      number_node = number_node + 1
86      string = string + ".ends\n"
87      file.write(string)

```

```

89 number_sub = 5
90
91 def decoder_3to8(file):
92     global number_node
93     global number_sub
94     string = ".subckt decoder_3to8"
95     str1 = ""
96     for i in range(3):
97         str1 = str1 + "\tin" + str(i) + "\t"
98     string = string + str1
99     str1 = ""
100    for i in range(8):
101        str1 = str1 + "\tout" + str(i)
102    string = string + str1 + "\ten\n"
103    str1 = ""
104    for i in range(3):
105        str1 = str1 + "x" + str(number_sub) + "\tin" + str(i) + "\t" + "n" + str(number_node) + "\tinverter\n"
106        number_sub = number_sub + 1
107        number_node = number_node + 1
108    number_node = number_node - 3
109    string = string + str1
110    str1 = ""
111    for i in range(8):
112        str1 = str1 + "x" + str(number_sub) + "\t" + ("n" + str(number_node) if i%2 == 0 else "in" + str(0)) \
113        + ( "\tn" + str(number_node+1) if (int)(i/2)%2 == 0 else "\tin" + str(1) ) \
114        + ( "\tn" + str(number_node+2) if (int)(i/4)%2 == 0 else "\tin" + str(2) ) + "\ten\t" + "out" + str(i) + "\tand_4\n"
115        number_sub = number_sub + 1
116    string = string + str1
117
118    string = string + ".ends\n\n"
119    file.write(string)

```

```

121 def decoder_6to64(file):
122     global number_sub
123     string = ".subckt decoder_6to64"
124     str1 = ""
125     for i in range(6):
126         str1 = str1 + "\tin" + str(i) + "\t"
127     string = string + str1
128     str1 = ""
129     for i in range(64):
130         str1 = str1 + "\tout" + str(i)
131     string = string + str1 + "\ten\n"
132     str1 = ""
133     for i in range(8):
134         str1 = str1 + "x" + str(number_sub) + "\tin0\tin1\tin2" + "".join([ (" \tout" + str(i*8+j) ) for j in range(8) ]) + "\ten" + str(i) + "\tdecoder_3to8\n"
135         number_sub = number_sub + 1
136     string = string + str1
137     str1 = ""
138     for i in range(1):
139         str1 = str1 + "x" + str(number_sub) + "\tin3\tin4\tin5" + "".join([ (" \ten" + str(i*j) ) for j in range(8) ]) + "\ten\tdecoder_3to8\n"
140         number_sub = number_sub + 1
141     string = string + str1
142
143     string = string + ".ends\n\n"
144     file.write(string)
145
146 def init(file,num_addr):
147     file.write(".TITLE Decoder\n\n")
148     file.write(".protect\n\n")
149     file.write(".include 'RAID2/COURSE/mcs/mcs020/Technology_files/7nm_tt.pm'\n\n")
150     file.write(".unprotect\n\n")
151     file.write(".vec 'input.vec'\n\n")
152     file.write(".global VDD GND\n\n")
153     #NAND_2(file)
154     INV(file)
155     NAND(file,num_addr)
156     AND(file,num_addr)
157     decoder_3to8(file)
158     decoder_6to64(file)

```

```

161 def circuit_construct(file):
162     string = ""
163     str1 = ""
164     for i in range(6):
165         str1 = str1 + "Xin" + str(i) + "\tAddr" + "[" + str(i) + "]" + "\tin_i" + "[" + str(i) + "]" + "\tin_v_chain\n"
166     string = string + str1 + "\n"
167
168     str1 = ""
169     for i in range(1):
170         str1 = str1 + "Xx" + str(i) + "\t" + "".join([ (" \tin_i" + "[" + str(j) + "]" ) for j in range(6) ]) \
171         + "".join([ (" \tWL" + "[" + str(j) + "]" ) for j in range(64) ]) + "\tclk\tdecoder_6to64\n"
172     string = string + str1 + "\n"
173
174     str1 = ""
175     for i in range(64):
176         str1 = str1 + "C" + str(i) + "\tWL" + "[" + str(i) + "]" + "\t" + "GND\t" + "22.2592a\n"
177     string = string + str1 + "\n"
178
179     file.write(string)

```

```

181
182 #####
183 #-----main-----#
184 #####
185
186 fo = open("decoder_6to64.sp", "w")
187 vector()
188 init(fo,num_addr)
189 circuit_construct(fo)
190 fo.write("Vvdd VDD GND dc 0.7v\n\n\
191         ".op\n\n\
192         ".option post      \n\n\
193         ".options probe    \n\n\
194         ".probe v(*) i(*) \n\n\
195         ".option captab    \n\n\
196         ".tran 0.01ns 70ns\n\n")
197 fo.write(".measure TRAN pwr avg power\n\n\
198         ".TEMP 25\n\n\
199         ".end\n\n")
200 fo.close()

```

如以上程式碼所示，使用 python 來產生.sp 檔案並產生 input.vec 檔。

設定 $VDD = 0.7V$ ，Frequency = 1GHz，duty cycle = 50%，rise time & fall time = 0.05ns，並在 output 接上 22.2592aF 的電容，其值透過 6T-SRAM 的 gate capacitances of the pass transistors 來找，如下圖。

decoder_6to64.lis

read_111.lis X

write_111.lis

decoder_6to64.sp

lab1 > read_111.lis

841***** HSPICE -- Q-2020.03-SP2-2 linux64 (Nov 1 2020) *****

120

121

122**** mosfets

123

124

125subckt

126element0:mpr0:mnr0:mpl0:mn10:mnpr0:mnpl

127model0:pmos_sra0:nmos_sra0:pmos_sra0:nmos_sra0:nmos_sra0:nmos_sra

128regionSubthSubthSubthSubthSubthSubth

129id-448.4133n564.6288n-448.4133n564.6288n116.2155n116.2155n

130ibs0.0.0.0.0.0.

131ibd0.0.0.0.0.0.

132vgs-326.7236m373.2764m-326.7236m373.2764m326.7236m326.7236m

133vds-326.7236m373.2764m-326.7236m373.2764m326.7236m326.7236m

134vbs-110.6381m589.3619m-110.6381m589.3619m216.0854m216.0854m

135vth-334.8227m408.4753m-334.8227m408.4753m408.7317m408.7317m

136vdsat-93.7528m80.0655m-93.7528m80.0655m72.9074m72.9074m

137vod8.0992m-35.1988m8.0992m-35.1988m-82.0081m-82.0081m

138beta1.6413m2.1736m1.6413m2.1736m2.1736m2.1736m

139gam_eff0.0.0.0.0.0.

140gm11.9719u17.0437u11.9719u17.0437u4.1690u4.1690u

141gds276.9647n265.8642n276.9647n265.8642n66.2844n66.2844n

142gmb4.6836p11.8438p4.6836p11.8438p1.4851p1.4851p

143cdtot10.1770a10.2207a10.1770a10.2207a10.2213a10.2213a

144cgtot30.7262a26.6272a30.7262a26.6272a22.2592a22.2592a

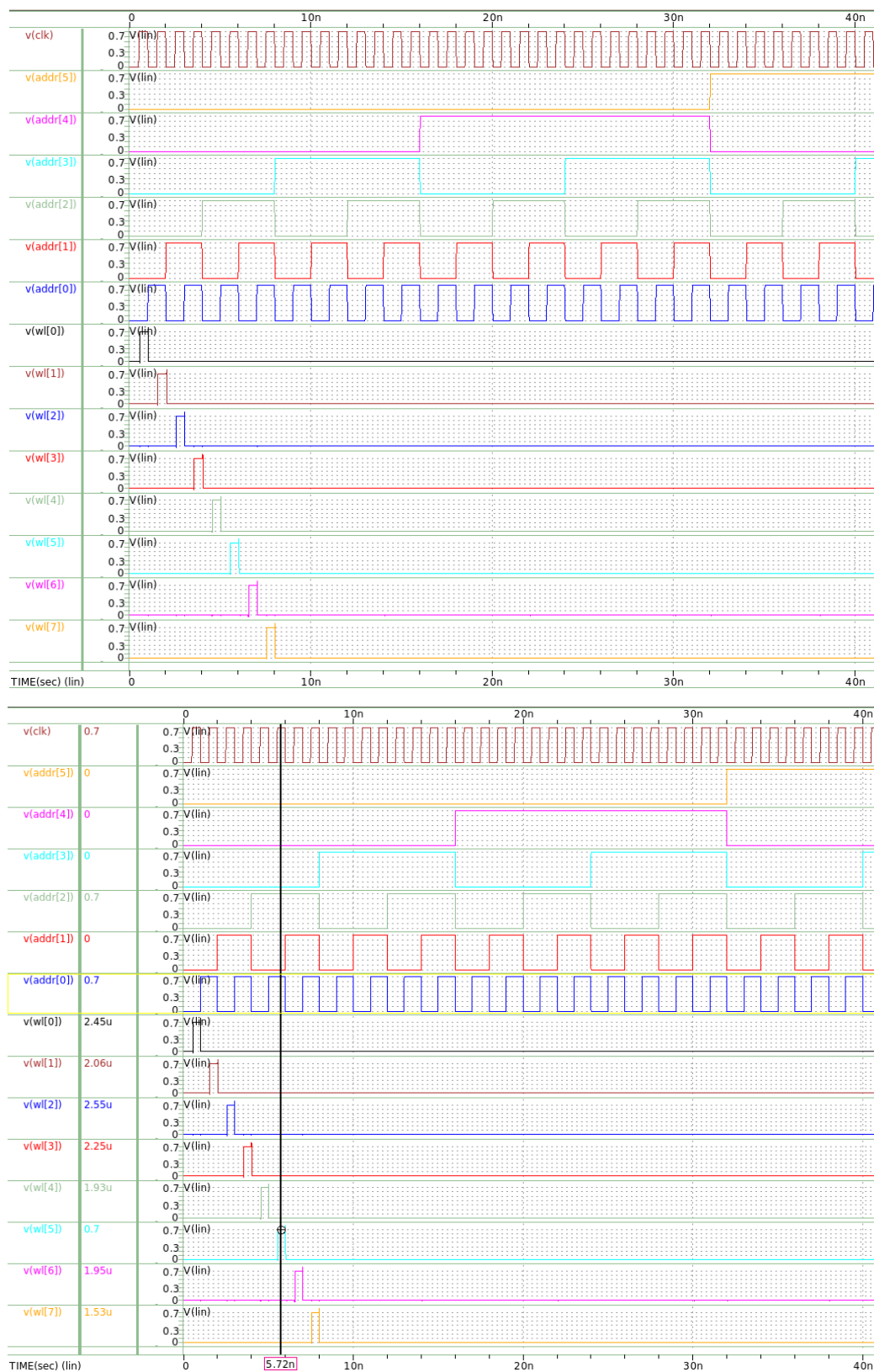
145cstot16.9362a14.3868a16.9362a14.3868a11.4589a11.4589a

146cbtot0.0.0.0.0.0.

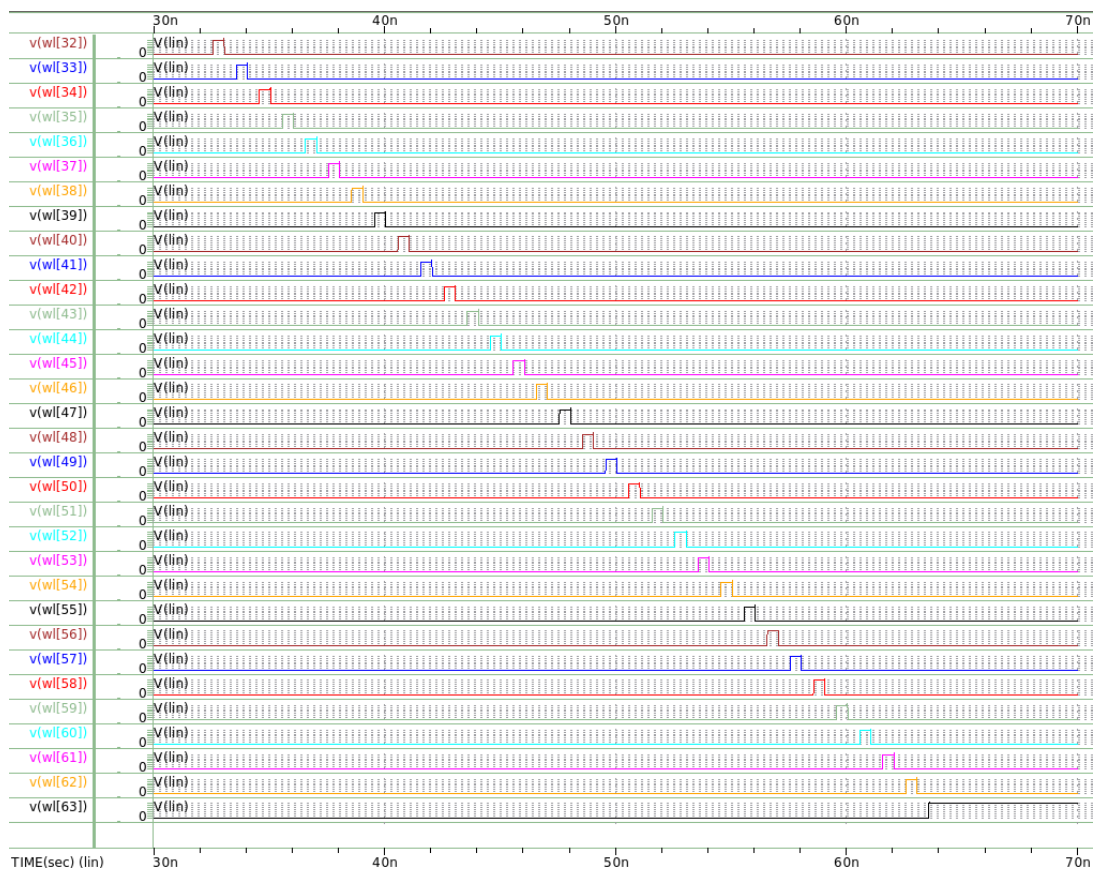
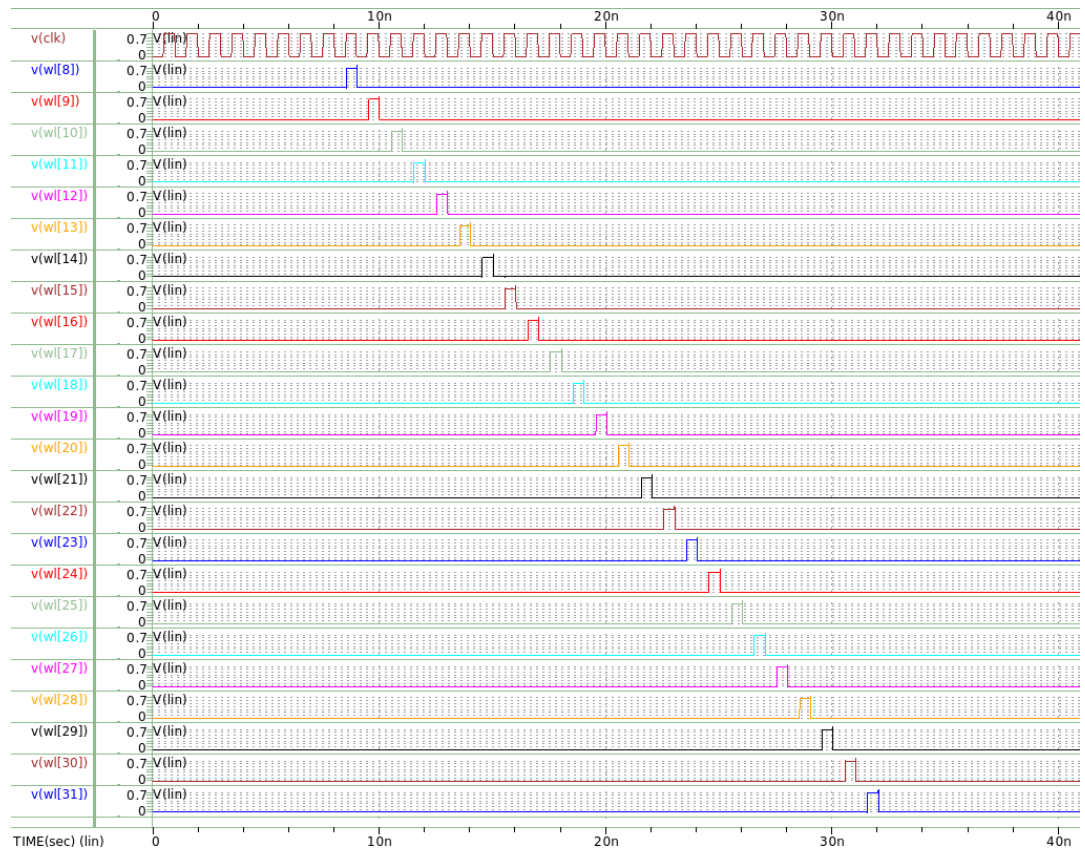
147cgs20.6543a16.4276a20.6543a16.4276a12.0453a12.0453a

148cgd10.0719a10.1996a10.0719a10.1996a10.2138a10.2138a

149

Result :

如上圖所示，以 wl[5] 來看，add[5:0] = 000101，clk = 0.7 (high) 時，wl[5] = 0.7 (high)，表示有成功 decoder，也順利達到 clk 為 high 時 wordline 會拉起來的 SPEC.。



以上為 wl[0] to wl[63]的結果。

Performance :

在表現上可以發現，wordline 都有正常運作，但會發現在拉起為 1 時，會有 Gibbs phenomenon，可能的原因是在 address 切換的時候，會讓訊號不穩，或是電容充放電的效應。

Power :

```
1 $DATA1 SOURCE='HSPICE' VERSION='Q-2020.03-SP2-2 linux64' PARAM_COUNT=0
2 .TITLE '.title decoder'
3 pwr          temper          alter#
4 3.026e-06     25.0000         1
5
```