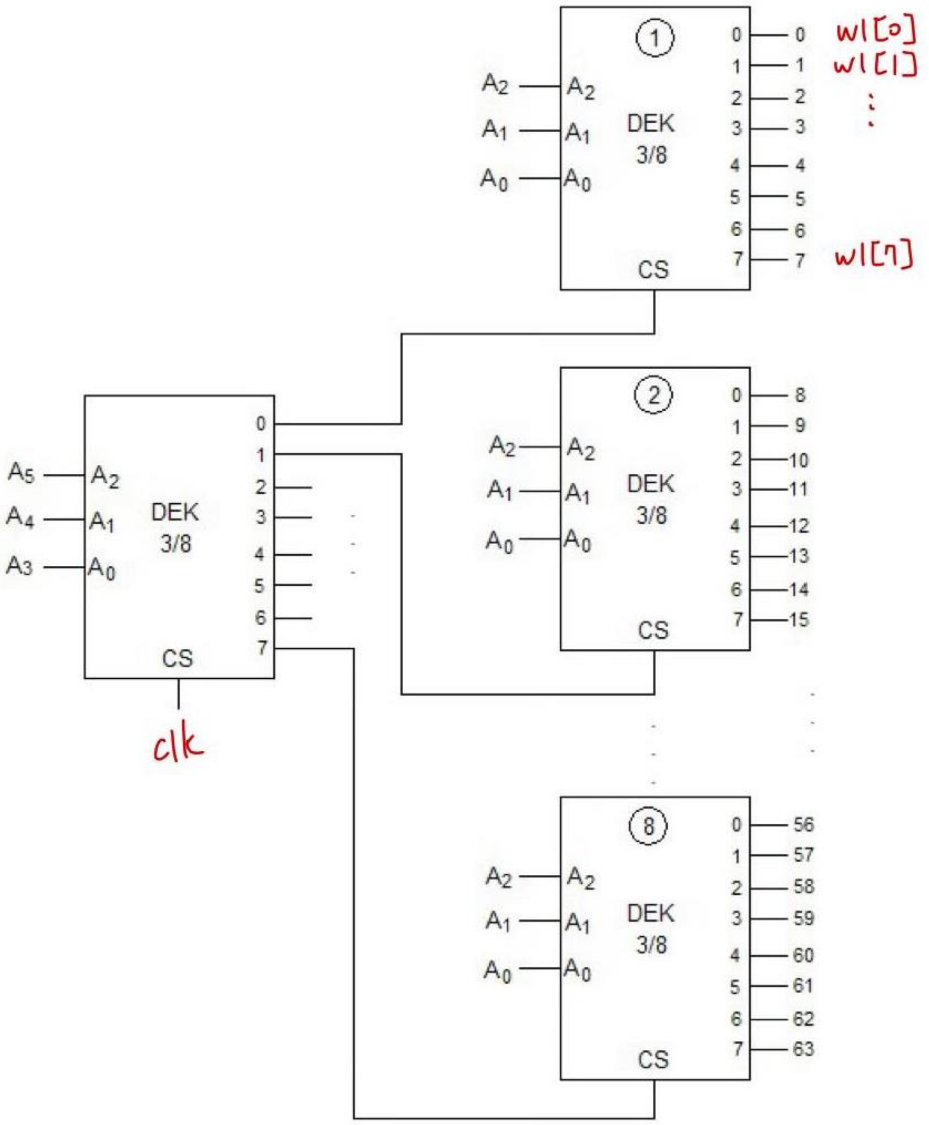
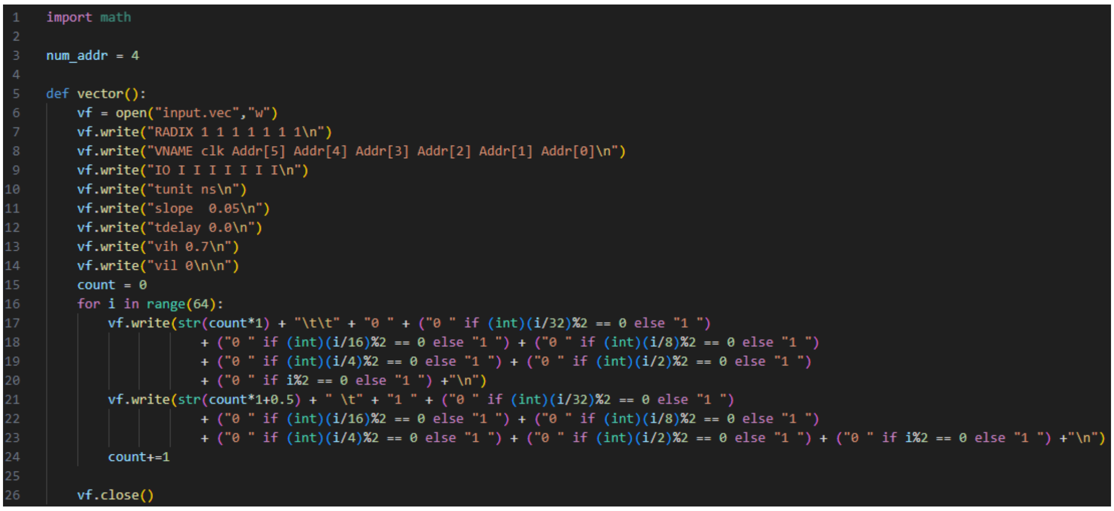
Memory Circuits and System Report HW2

Architecture：

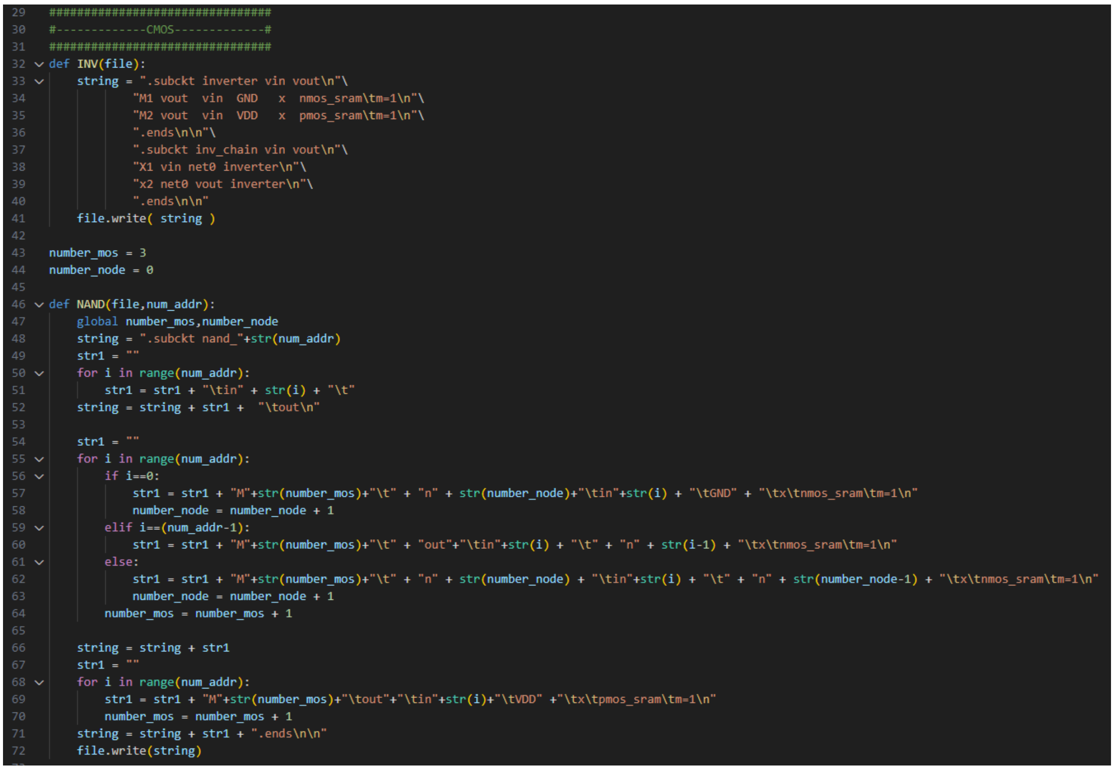


使用 9 個 3to8_decoder 來設計 6to64_decoder，將其中一個的輸出接上剩下 8 個的 enable，如此便能控制 WL 輸出，再將 clk 接上 MSB 端 decoder 的 enable 來達到 decoder 功用。

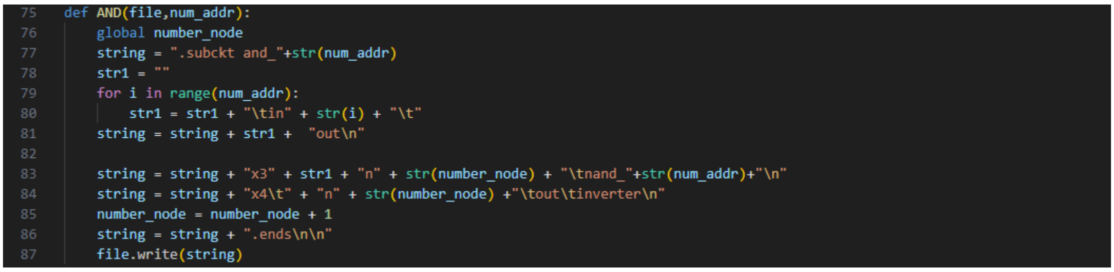
Python：



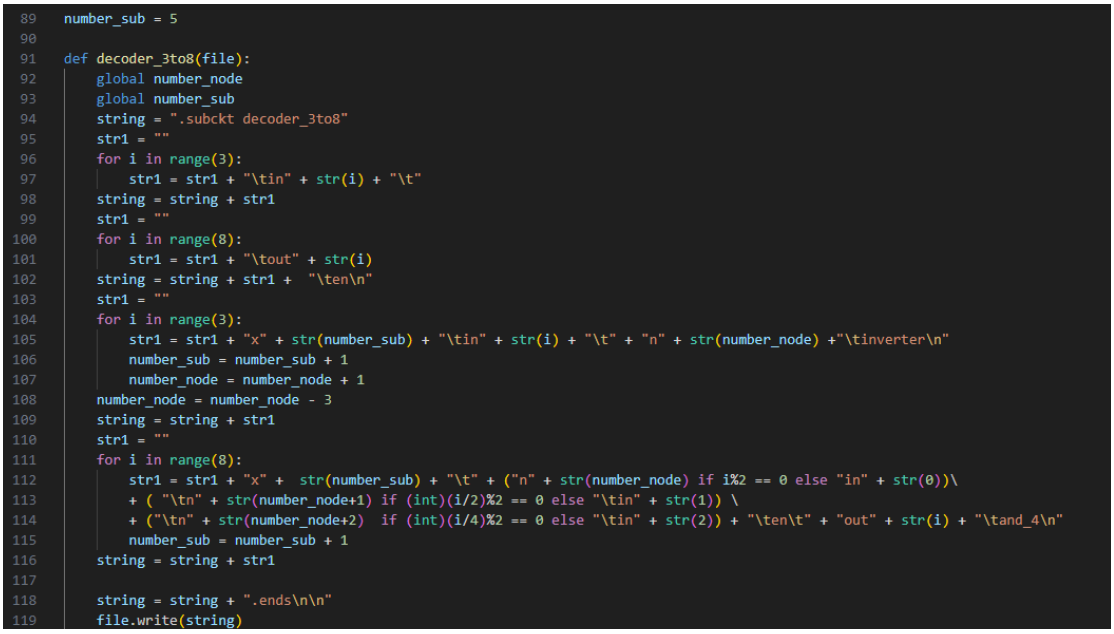
```python
import math

num_addr = 4

def vector():
    vf = open("input.vec","w")
    vf.write("RADIX 1 1 1 1 1 1 1\n")
    vf.write("VNAME clk Addr[5] Addr[4] Addr[3] Addr[2] Addr[1] Addr[0]\n")
    vf.write("IO I I I I I I I\n")
    vf.write("tunit ns\n")
    vf.write("slope  0.05\n")
    vf.write("tdelay 0.0\n")
    vf.write("vih 0.7\n")
    vf.write("vil 0\n\n")
    count = 0
    for i in range(64):
        vf.write(str(count*1) + "\t\t" + "0 " + ("0 " if (int)(i/32)%2 == 0 else "1 ")
                    + ("0 " if (int)(i/16)%2 == 0 else "1 ") + ("0 " if (int)(i/8)%2 == 0 else "1 ")
                    + ("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")
        vf.write(str(count*1+0.5) + " \t" + "1 " + ("0 " if (int)(i/32)%2 == 0 else "1 ")
                    + ("0 " if (int)(i/16)%2 == 0 else "1 ") + ("0 " if (int)(i/8)%2 == 0 else "1 ")
                    + ("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")
        count+=1

    vf.close()
```



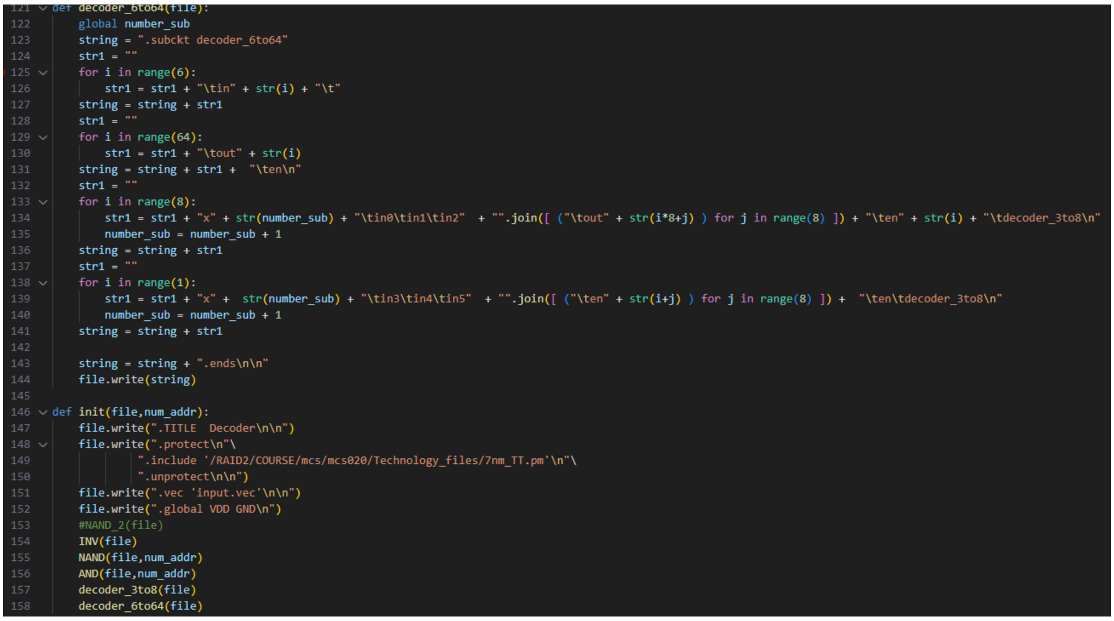
```python
################################
#------------CMOS------------#
################################
def INV(file):
    string = ".subckt inverter vin vout\n"\
            "M1 vout  vin  GND   x  nmos_sram\tm=1\n"\
            "M2 vout  vin  VDD   x  pmos_sram\tm=1\n"\
            ".ends\n\n"\
            ".subckt inv_chain vin vout\n"\
            "X1 vin net0 inverter\n"\
            "x2 net0 vout inverter\n"\
            ".ends\n\n"
    file.write( string )

number_mos = 3
number_node = 0

def NAND(file,num_addr):
    global number_mos,number_node
    string = ".subckt nand_"+str(num_addr)
    str1 = ""
    for i in range(num_addr):
        str1 = str1 + "\tin" + str(i) + "\t"
    string = string + str1 +  "\tout\n"

    str1 = ""
    for i in range(num_addr):
        if i==0:
            str1 = str1 + "M"+str(number_mos)+"\t" + "n" + str(number_node)+"\tin"+str(i) + "\tGND" + "\tx\tnmos_sram\tm=1\n"
            number_node = number_node + 1
        elif i==(num_addr-1):
            str1 = str1 + "M"+str(number_mos)+"\t" + "out"+"\tin"+str(i) + "\t" + "n" + str(i-1) + "\tx\tnmos_sram\tm=1\n"
        else:
            str1 = str1 + "M"+str(number_mos)+"\t" + "n" + str(number_node) + "\tin"+str(i) + "\t" + "n" + str(number_node-1) + "\tx\tnmos_sram\tm=1\n"
            number_node = number_node + 1
        number_mos = number_mos + 1

    string = string + str1
    str1 = ""
    for i in range(num_addr):
        str1 = str1 + "M"+str(number_mos)+"\tout"+"\tin"+str(i)+"\tVDD" +"\tx\tpmos_sram\tm=1\n"
        number_mos = number_mos + 1
    string = string + str1 + ".ends\n\n"
    file.write(string)
```



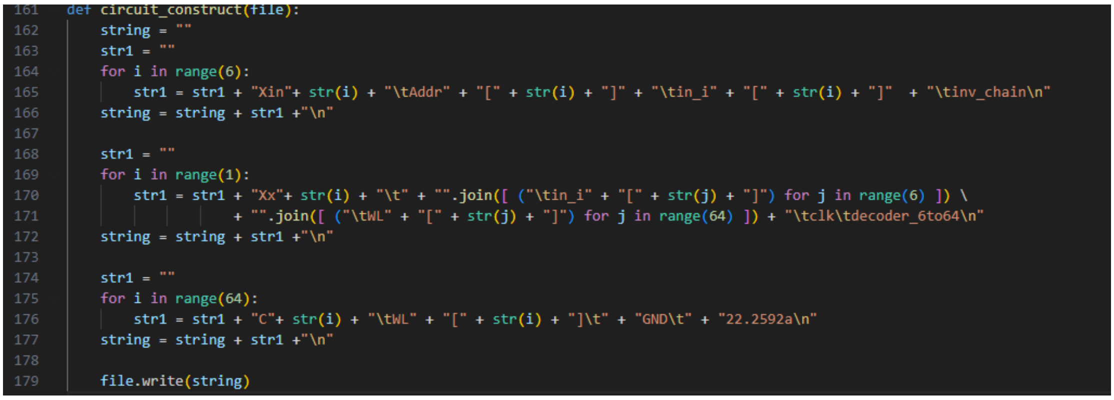
```python
def AND(file,num_addr):
    global number_node
    string = ".subckt and_"+str(num_addr)
    str1 = ""
    for i in range(num_addr):
        str1 = str1 + "\tin" + str(i) + "\t"
    string = string + str1 +  "out\n"

    string = string + "x3" + str1 + "n" + str(number_node) + "\tnand_"+str(num_addr)+"\n"
    string = string + "x4\t" + "n" + str(number_node) +"\tout\tinverter\n"
    number_node = number_node + 1
    string = string + ".ends\n\n"
    file.write(string)
```



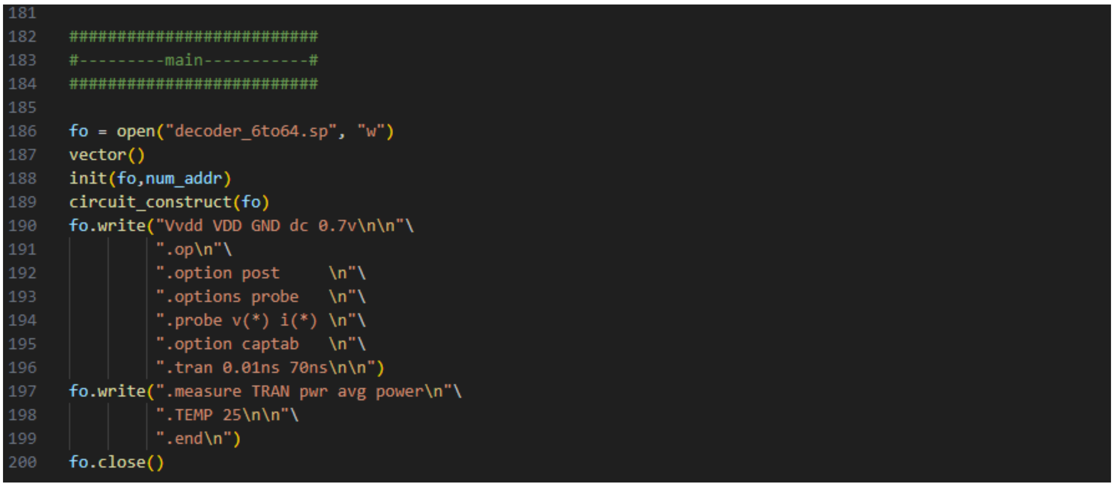
```python
number_sub = 5

def decoder_3to8(file):
    global number_node
    global number_sub
    string = ".subckt decoder_3to8"
    str1 = ""
    for i in range(3):
        str1 = str1 + "\tin" + str(i) + "\t"
    string = string + str1
    str1 = ""
    for i in range(8):
        str1 = str1 + "\tout" + str(i)
    string = string + str1 +  "\ten\n"
    str1 = ""
    for i in range(3):
        str1 = str1 + "x" + str(number_sub) + "\tin" + str(i) + "\t" + "n" + str(number_node) +"\tinverter\n"
        number_sub = number_sub + 1
        number_node = number_node + 1
    number_node = number_node - 3
    string = string + str1
    str1 = ""
    for i in range(8):
        str1 = str1 + "x" +  str(number_sub) + "\t" + ("n" + str(number_node) if i%2 == 0 else "in" + str(0))\
        + ( "\tn" + str(number_node+1) if (int)(i/2)%2 == 0 else "\tin" + str(1)) \
        + ("\tn" + str(number_node+2)  if (int)(i/4)%2 == 0 else "\tin" + str(2)) + "\ten\t" + "out" + str(i) + "\tand_4\n"
        number_sub = number_sub + 1
    string = string + str1

    string = string + ".ends\n\n"
    file.write(string)
```



```python
def decoder_6to64(file):
    global number_sub
    string = ".subckt decoder_6to64"
    str1 = ""
    for i in range(6):
        str1 = str1 + "\tin" + str(i) + "\t"
    string = string + str1
    str1 = ""
    for i in range(64):
        str1 = str1 + "\tout" + str(i)
    string = string + str1 +  "\ten\n"
    str1 = ""
    for i in range(8):
        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"
        number_sub = number_sub + 1
    string = string + str1
    str1 = ""
    for i in range(1):
        str1 = str1 + "x" +  str(number_sub) + "\tin3\tin4\tin5"  + "".join([ ("\ten" + str(i+j) ) for j in range(8) ]) +  "\ten\tdecoder_3to8\n"
        number_sub = number_sub + 1
    string = string + str1

    string = string + ".ends\n\n"
    file.write(string)

def init(file,num_addr):
    file.write(".TITLE  Decoder\n\n")
    file.write(".protect\n"\
            ".include '/RAID2/COURSE/mcs/mcs020/Technology_files/7nm_TT.pm'\n"\
            ".unprotect\n\n")
    file.write(".vec 'input.vec'\n\n")
    file.write(".global VDD GND\n")
    #NAND_2(file)
    INV(file)
    NAND(file,num_addr)
    AND(file,num_addr)
    decoder_3to8(file)
    decoder_6to64(file)
```



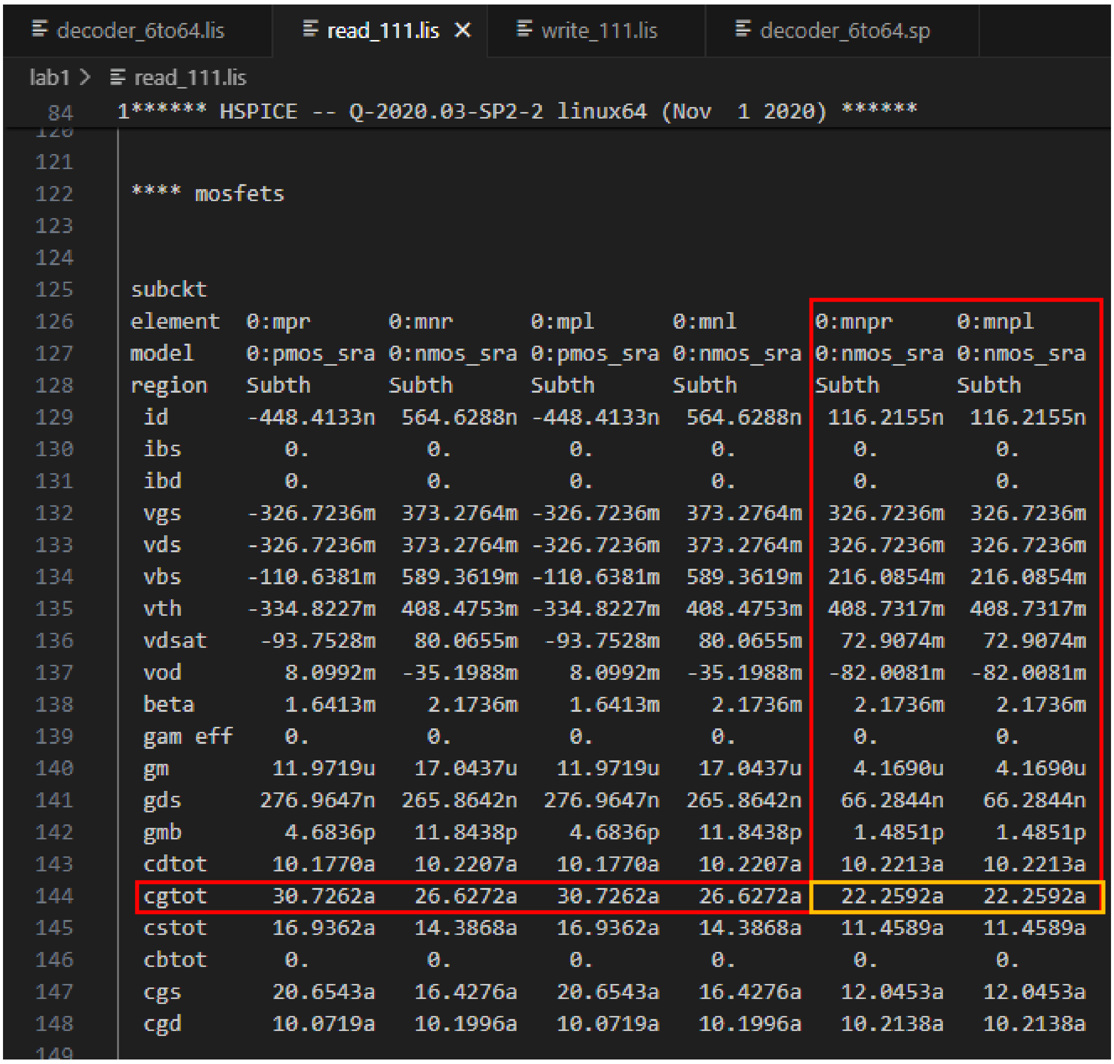
```python
def circuit_construct(file):
    string = ""
    str1 = ""
    for i in range(6):
        str1 = str1 + "Xin"+ str(i) + "\tAddr" + "[" + str(i) + "]" + "\tin_i" + "[" + str(i) + "]"  + "\tinv_chain\n"
    string = string + str1 +"\n"

    str1 = ""
    for i in range(1):
        str1 = str1 + "Xx"+ str(i) + "\t" + "".join([ ("\tin_i" + "[" + str(j) + "]") for j in range(6) ]) \
                    + "".join([ ("\tWL" + "[" + str(j) + "]") for j in range(64) ]) + "\tclk\tdecoder_6to64\n"
    string = string + str1 +"\n"

    str1 = ""
    for i in range(64):
        str1 = str1 + "C"+ str(i) + "\tWL" + "[" + str(i) + "]\t" + "GND\t" + "22.2592a\n"
    string = string + str1 +"\n"

    file.write(string)
```



```
##########################
#---------main----------#
##########################

fo = open("decoder_6to64.sp", "w")
vector()
init(fo,num_addr)
circuit_construct(fo)
fo.write("Vdd VDD GND dc 0.7v\n\n"\
            ".op\n"\
            ".option post     \n"\
            ".options probe   \n"\
            ".probe v(*) i(*) \n"\
            ".option captab   \n"\
            ".tran 0.01ns 70ns\n\n")
fo.write(".measure TRAN pwr avg power\n"\
            ".TEMP 25\n\n"\
            ".end\n")
fo.close()
```

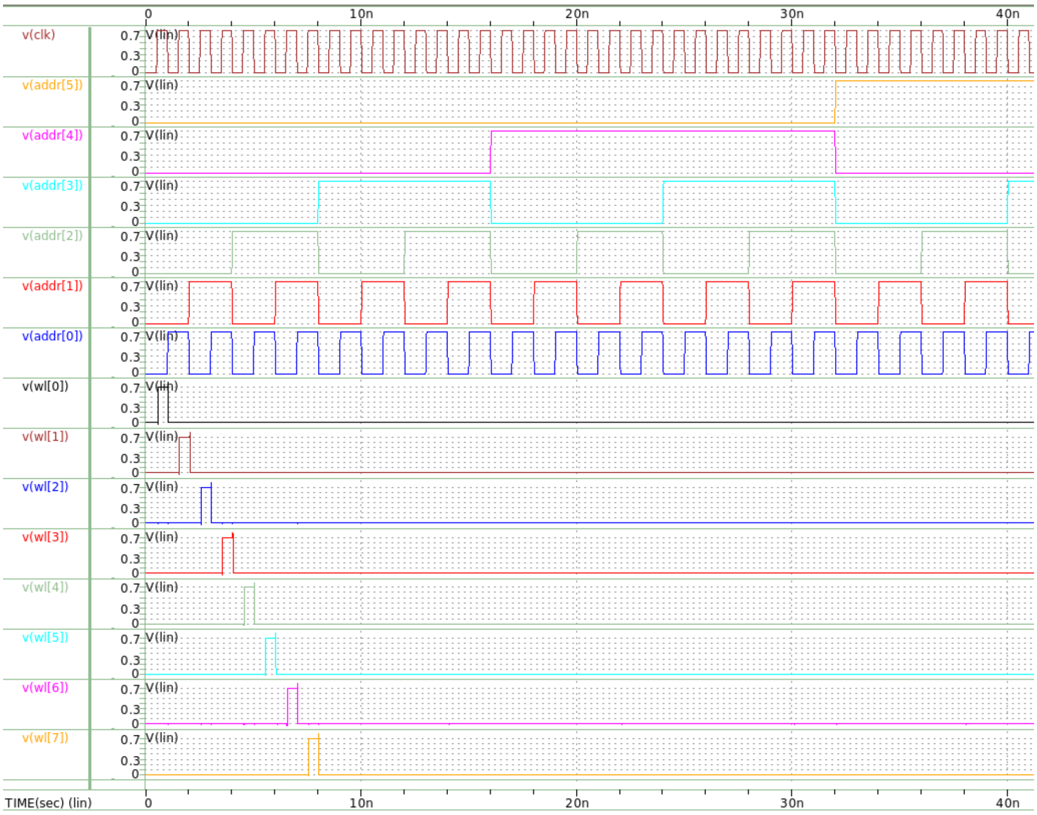
如以上程式碼所示，使用 python 來產生.sp 檔案並產生 input.vec 檔。

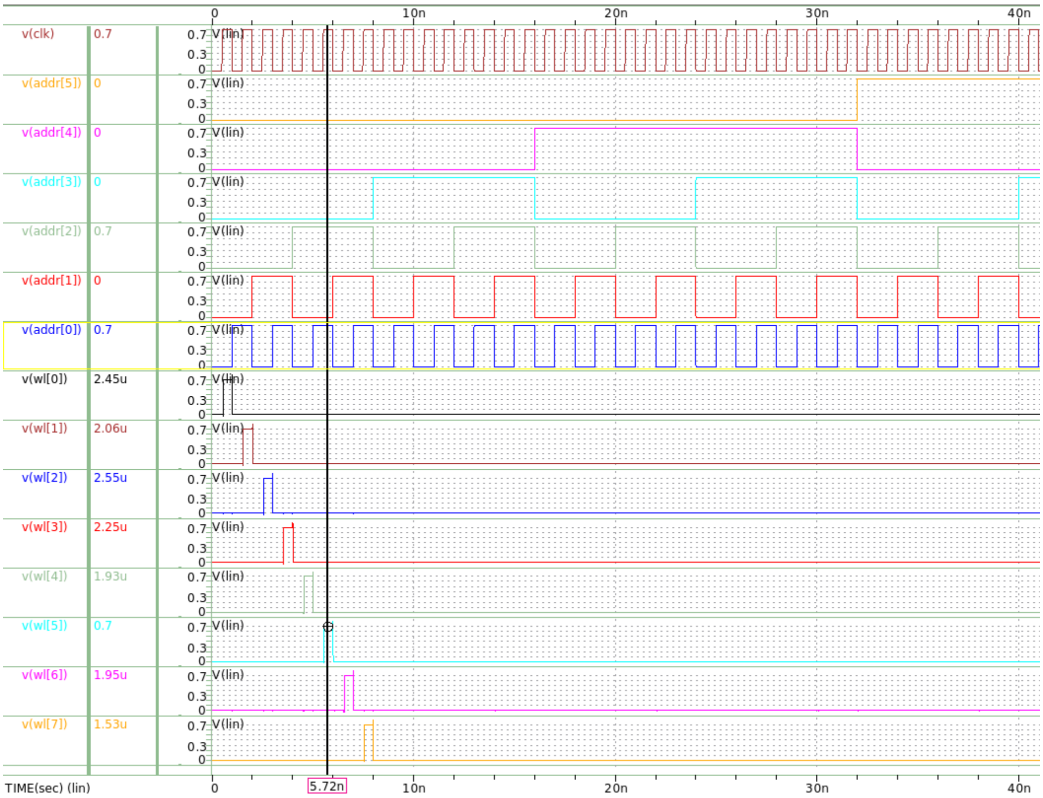
設定 VDD = 0.7V，Frequency = 1GHz，duty cycle = 50%，rise time & fall time = 0.05ns，並在 output 接上 22.2592aF 的電容，其值透過 6T-SRAM 的 gate capacitances of the pass transistors 來找，如下圖。



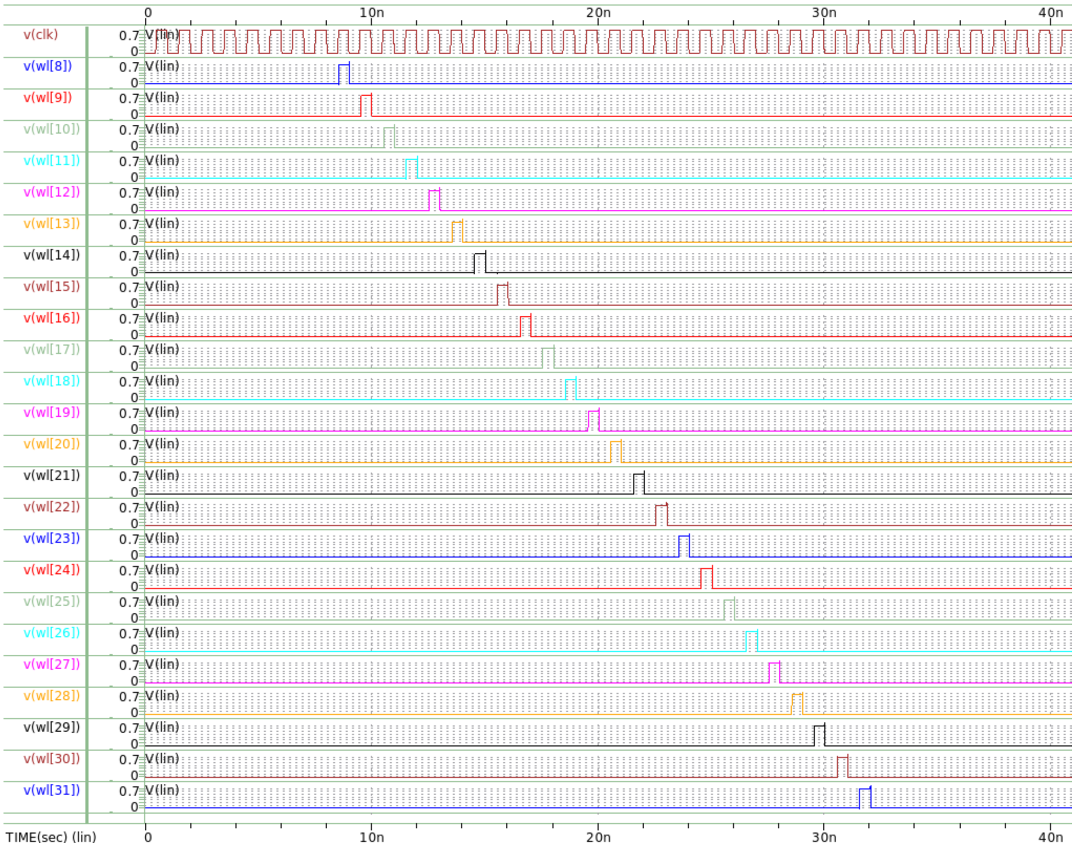
```
1****** HSPICE -- Q-2020.03-SP2-2 linux64 (Nov  1 2020) ******

 **** mosfets


 subckt
 element  0:mpr      0:mnr      0:mpl      0:mnl      0:mnpr     0:mnpl
 model    0:pmos_sra 0:nmos_sra 0:pmos_sra 0:nmos_sra 0:nmos_sra 0:nmos_sra
 region   Subth      Subth      Subth      Subth      Subth      Subth
  id      -448.4133n  564.6288n -448.4133n  564.6288n  116.2155n  116.2155n
  ibs        0.         0.         0.         0.         0.         0.
  ibd        0.         0.         0.         0.         0.         0.
  vgs     -326.7236m  373.2764m -326.7236m  373.2764m  326.7236m  326.7236m
  vds     -326.7236m  373.2764m -326.7236m  373.2764m  326.7236m  326.7236m
  vbs     -110.6381m  589.3619m -110.6381m  589.3619m  216.0854m  216.0854m
  vth     -334.8227m  408.4753m -334.8227m  408.4753m  408.7317m  408.7317m
  vdsat    -93.7528m   80.0655m  -93.7528m   80.0655m   72.9074m   72.9074m
  vod        8.0992m  -35.1988m    8.0992m  -35.1988m  -82.0081m  -82.0081m
  beta       1.6413m    2.1736m    1.6413m    2.1736m    2.1736m    2.1736m
  gam eff    0.         0.         0.         0.         0.         0.
  gm        11.9719u   17.0437u   11.9719u   17.0437u    4.1690u    4.1690u
  gds      276.9647n  265.8642n  276.9647n  265.8642n   66.2844n   66.2844n
  gmb        4.6836p   11.8438p    4.6836p   11.8438p    1.4851p    1.4851p
  cdtot     10.1770a   10.2207a   10.1770a   10.2207a   10.2213a   10.2213a
  cgtot     30.7262a   26.6272a   30.7262a   26.6272a   22.2592a   22.2592a
  cstot     16.9362a   14.3868a   16.9362a   14.3868a   11.4589a   11.4589a
  cbtot      0.         0.         0.         0.         0.         0.
  cgs       20.6543a   16.4276a   20.6543a   16.4276a   12.0453a   12.0453a
  cgd       10.0719a   10.1996a   10.0719a   10.1996a   10.2138a   10.2138a
```

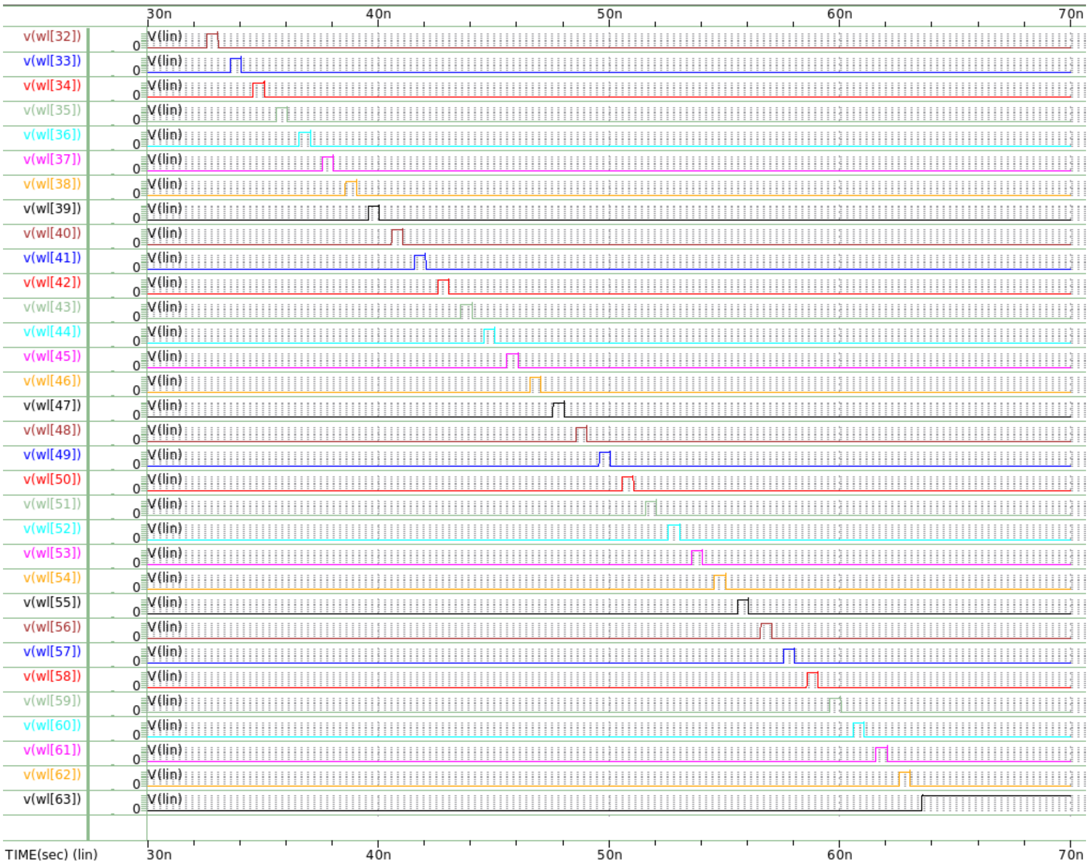
Result：





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.。



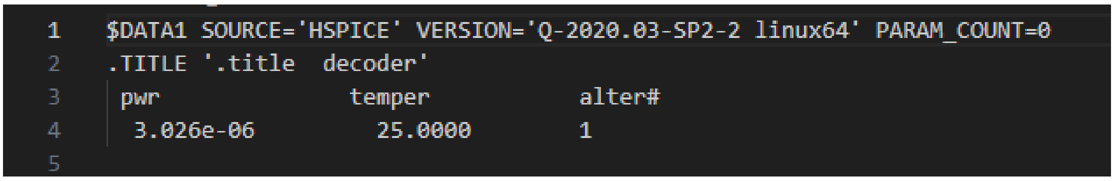


以上為 wl[0] to wl[63]的結果。

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：



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