

- (1) A 6T SRAM is shown in Fig. 1. Design the transistor widths for the SRAM to ensure proper functionality during both read and write operations. (20%) Generate the read and write operation curves for your SRAM cell, similar to those shown in Fig.2. (20%)

● Transistor widths in table format:

我的設計是根據老師上課講義所說

- P1 和 P2 最弱
- N2 和 N4 次強
- N1 和 N3 最強

考慮到 PMOS 的 mobility 比較低，所以 P1 和 P2 即使寬度為 64nm，也會比同樣是 64nm 寬的 N2 和 N4 弱。

No.	Width
P1	64nm
P2	64nm
N1	100nm
N2	64nm
N3	100nm
N4	64nm

此外，做 bitline conditioning 的兩個 PMOS 寬度如下:

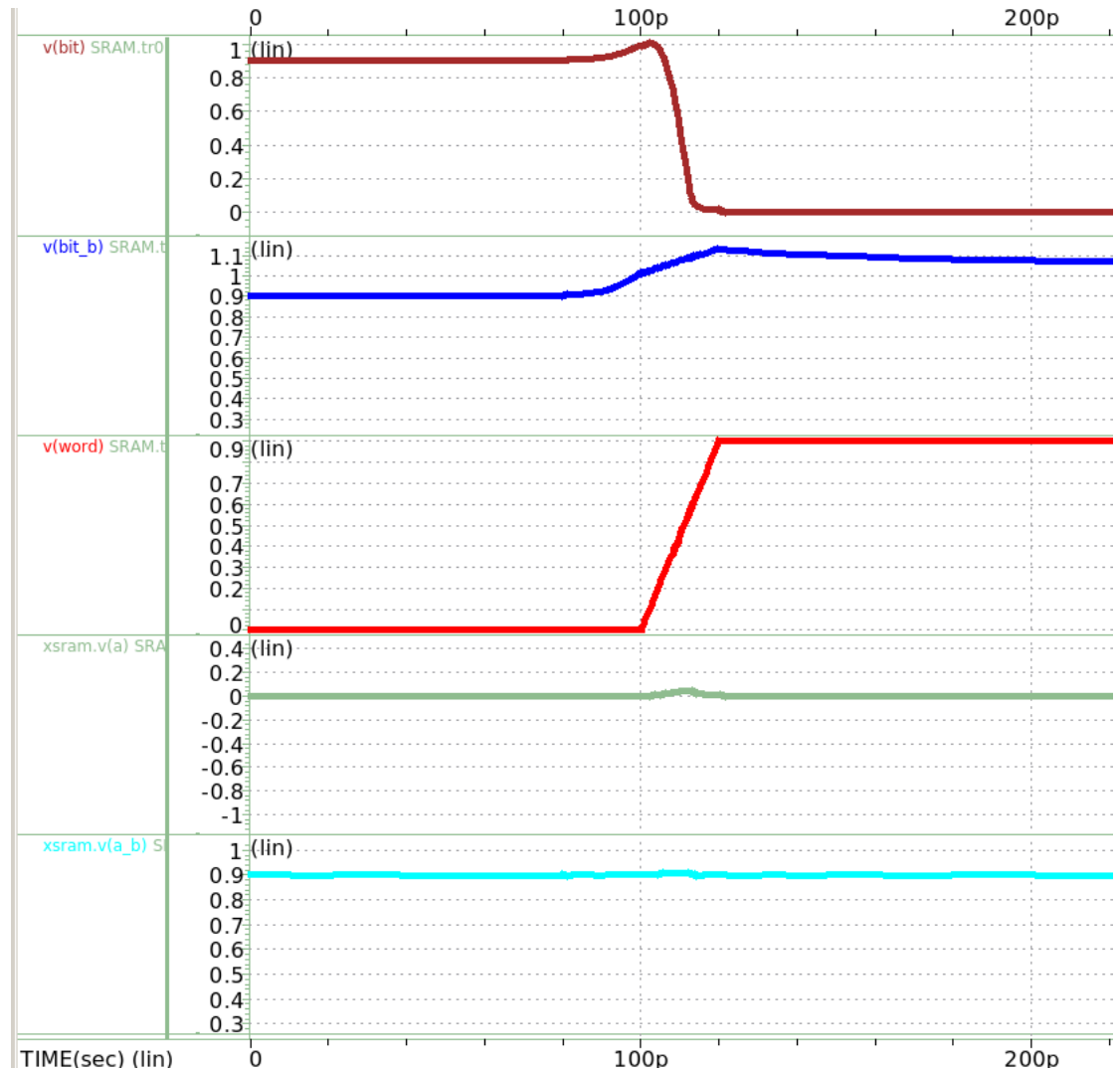
No.	Width
P1	64nm
P2	64nm

- Read Operation:

我有先把 A 的電壓初始化為 0。A_b 初始化為 VDD。

```
.ic v(xSRAM.A) = 0v
.ic v(xSRAM.A_b) = xvdd
```

下圖是我沒在 bit 及 bit_b 加 load 的波形圖:



從上圖可以看到我的 bit_b 高於 0.9v。

為了解決這個問題，所以我在 bit 及 bit_b 各加上 3fF 的電容:

```
c1 bit gnd! 3f
c2 bit_b gnd! 3f
```

The timing diagram displays five signals over a time interval from 0 to 200p. The signals are:

- v(bit) SRAM.tr0** (Red line): Starts at 0.9, remains constant until 100p, then decays to 0.1 by 200p.
- v(bit_b) SRAM.t** (Blue line): Starts at 0.9, remains constant until 100p, then jumps to 1.0 and remains constant.
- v(word) SRAM.t** (Red line): Starts at 0, remains constant until 100p, then jumps to 0.9 and remains constant.
- xsram.v(a) SRA** (Green line): Starts at 0, remains constant until 100p, then jumps to 0.2 and remains constant.
- xsram.v(a_b) SI** (Cyan line): Starts at 0.9, remains constant until 100p, then jumps to 1.0 and remains constant.

The diagram shows the timing of a word and bit signal relative to SRAM access times. The word signal (red) is a narrow pulse, while the bit signal (brown) is a wider pulse. The SRAM access times (green) are shown as a peak labeled 'A'. The bit signal is labeled 'bit_b' and the word signal is labeled 'word'. The SRAM access times are labeled 'A_b' and 'A'.

- Write Operation:

我先把 A 的電壓初始化為 0。A_b 初始化為 VDD。

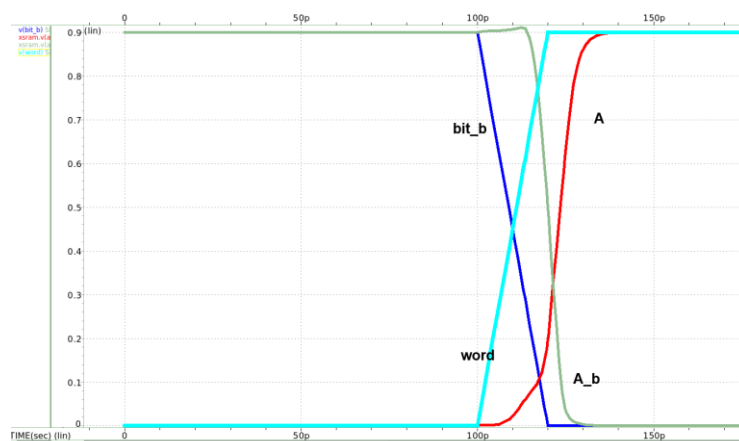
```
.ic v(xSRAM.A) = 0v
.ic v(xSRAM.A_b) = xvdd
```

接下來我試著寫入 A = 1。

下圖是我 write 的波形圖



疊圖後如下:



(2) Determine the butterfly curve and the quiescent Static Noise Margin (SVM_{hold}) of your SRAM cell in TT corners. (20%) Determine the butterfly curve and the read Static Noise Margin (SVM_{read}) of your SRAM cell TT corners. (20%)

- SVM_{hold} :

我是在 $word = 0$ 時跑模擬。

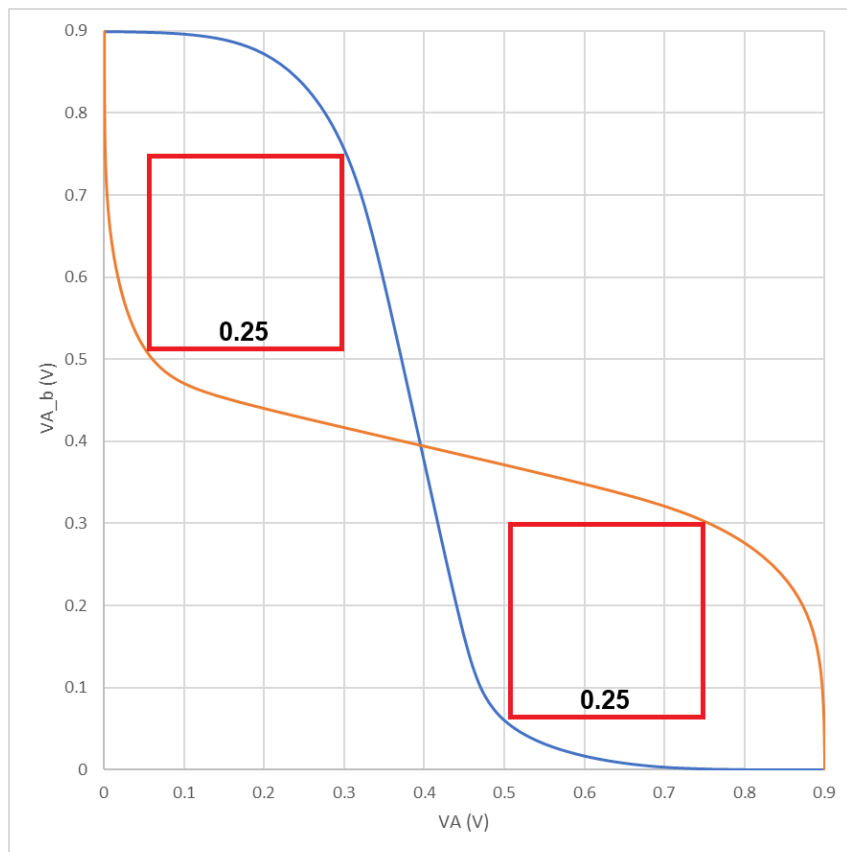
```
vphi phi gnd! xvdd
vw word gnd! 0v

* VA A gnd!
* .dc VA 0 xvdd 0.01
* .print V(A_b)

VA_b A_b gnd!
.dc VA_b 0 xvdd 0.01
.print V(A)
```

接著將兩組 VTC 數據丟進 excel 畫出下圖。

下圖是我模擬得出的 butterfly curve 加上 noise margin，正方形的邊長為 $0.25v$ 。



- SVM_{read}:

我把 bit 和 bit_b 都初始化為 VDD，然後 $\phi = VDD$ ，word = VDD
以模擬 read operation。

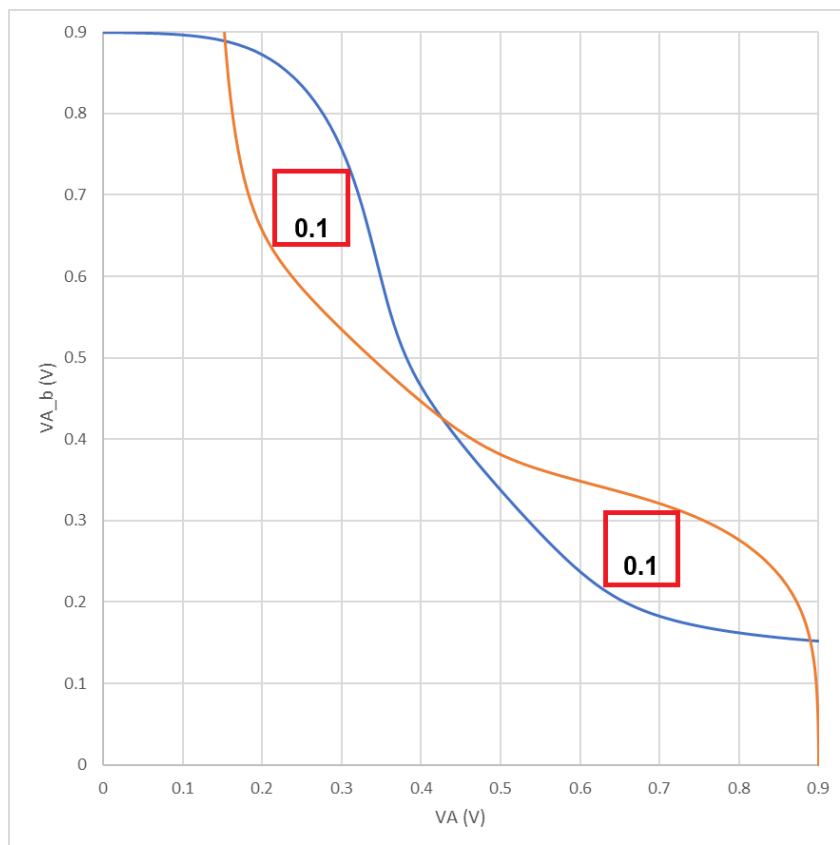
```
vphi phi gnd! xvdd
vw word gnd! xvdd
.ic V(bit) = xvdd
.ic V(bit_b) = xvdd

* VA A gnd!
* .dc VA 0 xvdd 0.01
* .print V(A_b)

VA_b A_b gnd!
.dc VA_b 0 xvdd 0.01
.print V(A)
```

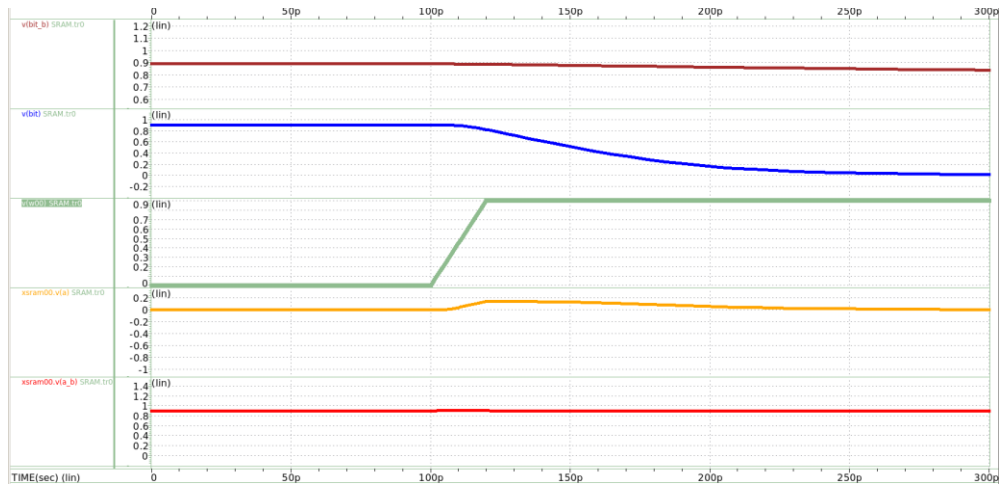
接著將兩組 VTC 數據都丟進 excel 畫出下圖。

下圖是我模擬得出的 butterfly curve 加上 noise margin，正方形的邊長為 $0.1v$ 。

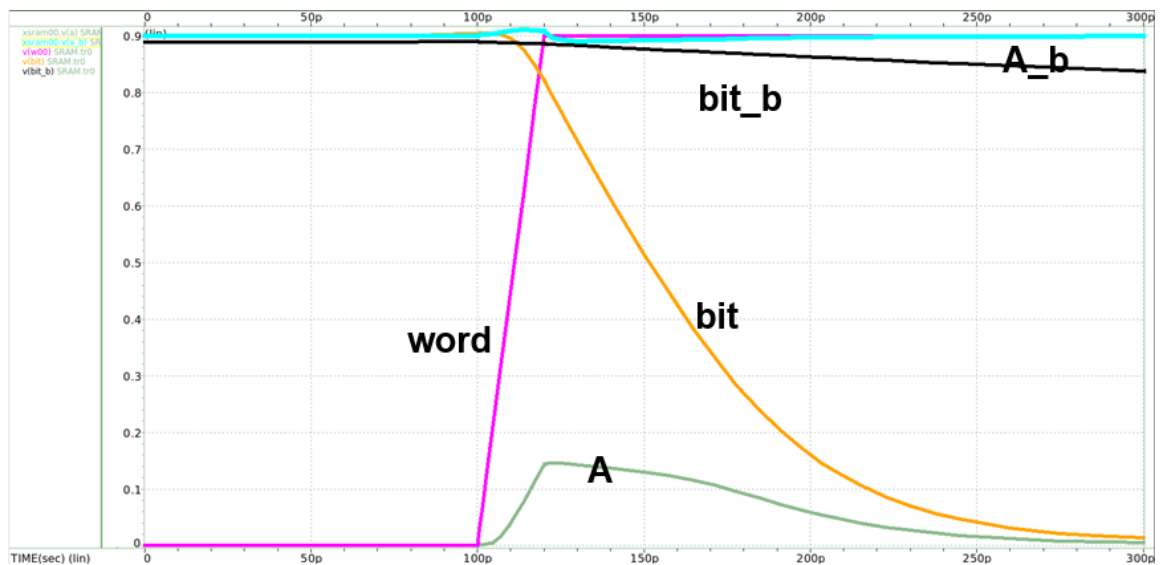


- (3) Design a 6T SRAM cell array under TT corner as shown in Fig. 3. Set the initial value of A=0 and A_b=1 in SRAM0. Set the initial value of A=1 and A_b=0 in SRAM₁~SRAM₆₃. Perform a proper read operation ($W_0 \sim W_{63}=0$, $\phi=0 \rightarrow \phi=1 \rightarrow W_0=1$) for SRAM₀. Generate the read operation curve and discuss the result to that of (1). What makes the difference? (20%)

下圖是 read 的波形圖:



疊圖後如下:



第一小題因為只有一個 SRAM，所以 bitline 上的 load 非常小，bitline 反而會因此有時候電壓略高過 VDD。

第三小題因為 SRAM 變多了，bit_b 上的 load 也變多了，所以 bit_b 的電壓變得比較低一些。此外，因為 load 變多的關係，bit_b 的電壓會緩慢往下掉，最終會穩定在 0.6v。但其實 bit_b 緩慢往下掉並不是大問題，因為實務上我們通常是看 bit 和 bit_b 哪邊先掉到 0，一旦其中一邊掉到 0 時我們就已經知道 SRAM 內部存的值了，而另一邊的電壓值是不是能穩定在 VDD 老實說並不重要。