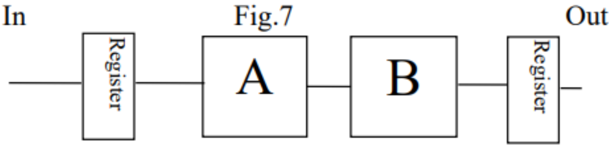
2023 Spring NYCU-EE Digital Integrated Circuits – Homework8

IEE 陳冠瑋 310510221 / May. 18, 2023

Consider the circuit in Fig.7. Modules A and B have a delay of 20 ns and 29.6 ns at 1.0 V (Vdd), and switch 20 pF and 35 pF (C) respectively. All the buses in Fig.7 are 20 bits. The register has a 0.4 ns delay and switches 0.05 pF. The clock rate of Fig.7 is thus 1/(50 ns) and the power dissipation is P0. The power dissipation can be estimated by P= C•Vdd^2 •F and the delay with respect to Vdd can be approximated by k/(Vdd-Vt) with Vt equals 0.3 V. k is a constant and is different for A and B. You can use the information of delay time, Vdd and Vt to calculate k.

Fig.7



題目提供的資訊整理如下:

Variable	Description	Value	Unit
T_A	Module A delay	20	ns
T_B	Module B delay	29.6	ns
T_{reg}	Register delay	0.4	ns
C_A	Capacitive load of A	20	pF
C_B	Capacitive load of B	35	pF
C_{reg}	Capacitive load of register (1 bit)	0.05	pF
Vdd	Vdd	1.0	V
Vt	Threshold voltage	0.3	V
F	Clock rate	20	MHz

由提供的資訊可以先計算出常數 K_A, K_B 與 P_0：

$$T_A = \frac{K_A}{V_{dd} - V_t} \rightarrow K_A = T_A \times (V_{dd} - V_t) = \ 20n \times (1.0 - 0.3) = \ 14n \ \ (\mathrm{V \cdot sec})$$

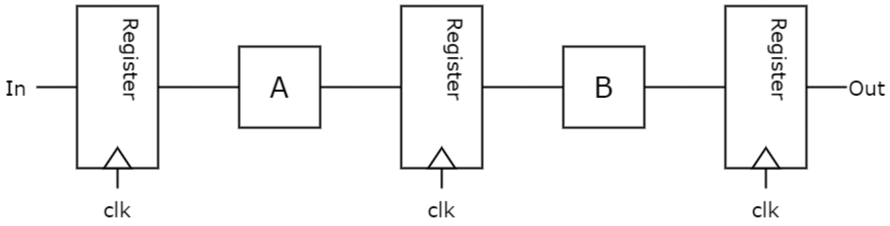
$$T_B = \frac{K_B}{V_{dd} - V_t} \rightarrow K_B = T_B \times (V_{dd} - V_t) = \ 29.6n \times (1.0 - 0.3) = \ 20.72n \ \ (\mathrm{V \cdot sec})$$

$$P_0 = C \times {V_{dd}}^2 \times F = \left(C_A + C_B + 2 \times C_{reg}\right) \times {V_{dd}}^2 \times F$$

$$\therefore P_0 = (20p + 35p + 2 \times 0.05p \times 20bits) \times 1.0^2 \times 20M = 1140 \ \ \mu\mathrm{W}$$

(a) Adding a pipeline register between A and B allows for reduction of the supply voltage (A and B can use different Vdd) while maintaining throughput (P1). Show the block diagram, explain the operation and calculate the power reduction ratio, P1/P0 (30%)

- **Block diagram**



- **Explain the operation**

利用 pipeline 的方法，在 module A 及 module B 中間加了一個 register，使 A、B 各自為一個 pipeline stage，在相同的 throughput 前提下就可以讓 clock cycle time 變長，讓 A 和 B 有將近 1 個 clock cycle 的時間運算，整體的 delay time 上升。

當整體的 delay time 變長，由$\text{delay time} = k/(\text{Vdd} - \text{Vt})$ 可知，Vdd 可以降低；當 Vdd 降低，由$P_{switch} = \text{C} \times Vdd^2 \times f$ 可知，switching power 會下降。

根據上述觀念，並考慮 register 本身的 delay 作為限制條件，可以計算出兩個 module 新的 Vdd:

$$\text{delay time} \leq 50 - 0.4 \text{ ns} = 49.6 \text{ ns}$$

$$T_A = 49.6 \ ns = \frac{14 \ ns}{V_{dd} - 0.3}, Vdd_A \approx 0.58 \ V$$

$$T_B = 49.6 \ ns = \frac{20.72 \ ns}{V_{dd} - 0.3}, Vdd_B \approx 0.72 \ V$$

根據以上算式，$Vdd_A = 0.58$ V，$Vdd_B = 0.72$ V。

- **Calculate the power ratio, P_1/P_0**

假設 Register 的 Vdd 維持在 1.0 V→ $Vdd_{reg} = 1.0$ V

前面已經先計算出 $P_0 = 1140 \ \mu\text{W}$

$$P_1 = C \times {V_{dd}}^2 \times F$$

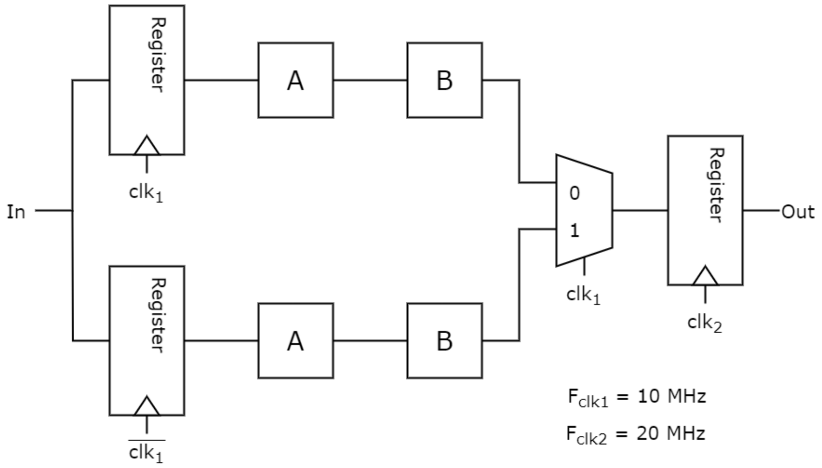
$$= 3 \times P_{reg} + P_A + P_B = (3 \times C_{reg} \times {Vdd_{reg}}^2 + C_A \times {Vdd_A}^2 + C_B \times {Vdd_B}^2) \times \text{F}$$

$$\therefore P_1 = (3 \times 0.05p \times 20 \times 1.0^2 + 20p \times 0.58^2 + 35p \times 0.72^2) \times 20\text{M} = 557.44 \ \mu\text{W}$$

$$\therefore \text{Power ratio} = \frac{P_1}{P_0} = \frac{557.44 \ \mu W}{1140 \ \mu W} \approx 48.9\%$$

(b) Assume that a 2-to-1 multiplexer has a delay of 0.4 ns at 1.0 V and switches 0.05 pF. Try a parallel version with two copies (using two A and B modules) while maintaining the data rate (P2). Show the block diagram, explain the operation, and calculate the power reduction ratio, P2/P0. (30%)

- **Block diagram**



- **Explain the operation**

運用 parallel 的觀念，將電路複製一份，兩份電路分別分成正緣觸發與負緣觸發，再經過一個 2-to-1 multiplexer 選擇 output 的結果，clk1 為 0 時選擇上面電路的結果，clk1 為 1 的時選擇下面電路的結果。此設計使電路再一個 clock cycle 裡共被觸發兩次，所以在相同的 throughput 的情況下，除了最後輸出的那一個 register 維持再 20 MHz，其餘的 register 頻率降為一半，也就是 10 MHz。

$$T_A + T_B = \frac{K_A}{Vdd_A - V_t} + \frac{K_B}{Vdd_B - V_t} \le 50 \times 2 - (0.4n + 0.4n)\ \ ns$$

$$\rightarrow \ \frac{14\ \ ns}{Vdd_A - 0.3} + \frac{20.72\ \ ns}{Vdd_B - 0.3} \le 99.2\ \ ns - (1)$$

$$P = P_A + P_B = \left(20p \times {Vdd_A}^2 + 35p \times {Vdd_B}^2\right) \times 10M - (2)$$

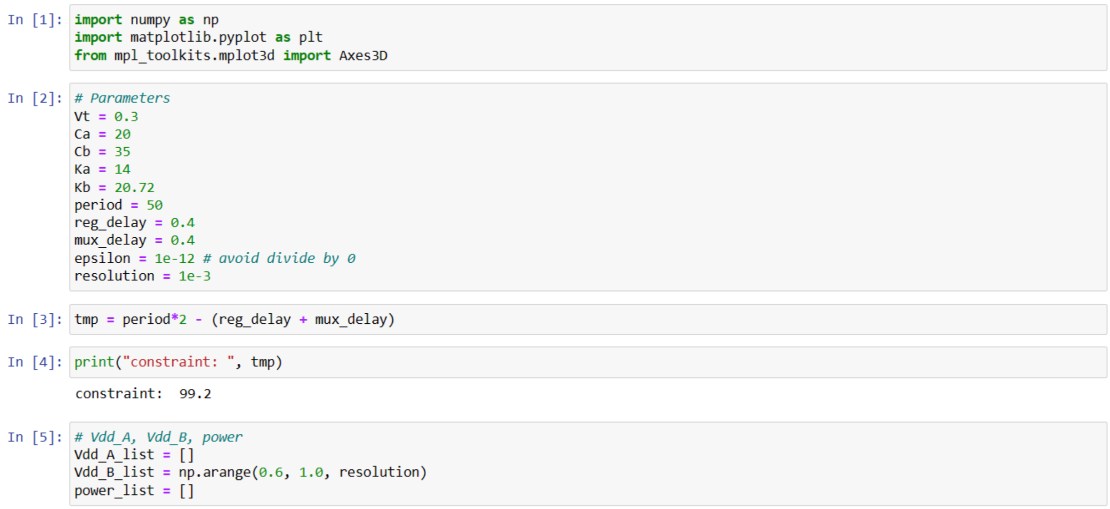
將(1)式Vdd_A代入(2)式:

- **Method 1:** 利用$\frac{dP}{dVdd_B} = 0$，求解極值

  可以得到$Vdd_B \approx 0.0276294\ \ or\ \ 0.64099\ \ \mathrm{V}$。
  由於$0.0276294 < \mathrm{Vt}$ 不合理，故取$Vdd_B = 0.641\ \ V$，代入(1)式可得到 $Vdd_A \approx 0.664$
  $P = P_A + P_B = (20p \times 0.664^2 + 35p \times 0.641^2) \times 10M = 231.98755\ \ \mu W$
  因此
  $Vdd_A = 0.664\ \ V$
  $Vdd_B = 0.641\ \ V$
  $P_A + P_B = 231.98755\ \ \mu W$

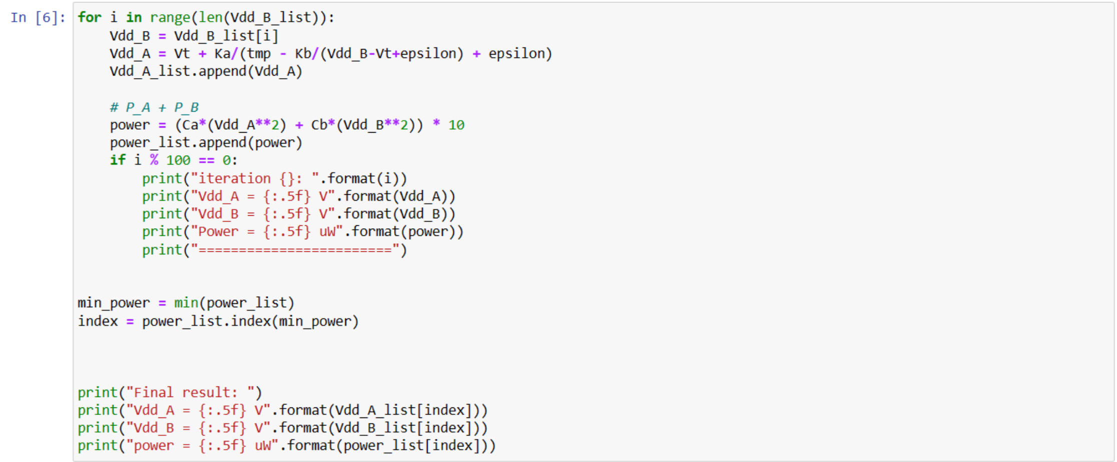
- **Method 2:** 利用迭代法求解 (python)

  在 P_A+P_B 最小值的情況下所得到的 $\mathrm{Vdd_A}$ 及 $\mathrm{Vdd_B}$ 即為最佳解。
  假設Vdd_B的範圍從 0.6 V 到 1.0V，且每次迭代的解析度為 0.001，將參數設定好，並將 Vdd 代入式子中:
  Note: epsilon = 1e-12 是為了避免分母為 0 而假設的一個很小的數值



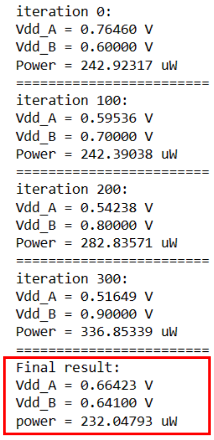
```
In [1]: import numpy as np
        import matplotlib.pyplot as plt
        from mpl_toolkits.mplot3d import Axes3D

In [2]: # Parameters
        Vt = 0.3
        Ca = 20
        Cb = 35
        Ka = 14
        Kb = 20.72
        period = 50
        reg_delay = 0.4
        mux_delay = 0.4
        epsilon = 1e-12 # avoid divide by 0
        resolution = 1e-3

In [3]: tmp = period*2 - (reg_delay + mux_delay)

In [4]: print("constraint: ", tmp)

        constraint:  99.2

In [5]: # Vdd_A, Vdd_B, power
        Vdd_A_list = []
        Vdd_B_list = np.arange(0.6, 1.0, resolution)
        power_list = []
```

將Vdd_B代入式子中，並計算Vdd_A和 power。最後，尋找整個迭代過程中 power 最小的值及對應的Vdd_A和Vdd_B。



```
In [6]: for i in range(len(Vdd_B_list)):
            Vdd_B = Vdd_B_list[i]
            Vdd_A = Vt + Ka/(tmp - Kb/(Vdd_B-Vt+epsilon) + epsilon)
            Vdd_A_list.append(Vdd_A)

            # P_A + P_B
            power = (Ca*(Vdd_A**2) + Cb*(Vdd_B**2)) * 10
            power_list.append(power)
            if i % 100 == 0:
                print("iteration {}: ".format(i))
                print("Vdd_A = {:.5f} V".format(Vdd_A))
                print("Vdd_B = {:.5f} V".format(Vdd_B))
                print("Power = {:.5f} uW".format(power))
                print("========================")


        min_power = min(power_list)
        index = power_list.index(min_power)


        print("Final result: ")
        print("Vdd_A = {:.5f} V".format(Vdd_A_list[index]))
        print("Vdd_B = {:.5f} V".format(Vdd_B_list[index]))
        print("power = {:.5f} uW".format(power_list[index]))
```

迭代的結果:



```
iteration 0:
Vdd_A = 0.76460 V
Vdd_B = 0.60000 V
Power = 242.92317 uW
========================
iteration 100:
Vdd_A = 0.59536 V
Vdd_B = 0.70000 V
Power = 242.39038 uW
========================
iteration 200:
Vdd_A = 0.54238 V
Vdd_B = 0.80000 V
Power = 282.83571 uW
========================
iteration 300:
Vdd_A = 0.51649 V
Vdd_B = 0.90000 V
Power = 336.85339 uW
========================
Final result:
Vdd_A = 0.66423 V
Vdd_B = 0.64100 V
power = 232.04793 uW
```

與 Method1 比較

- **Method 1:** 利用$\frac{dP}{dVdd_B} = 0$，求解極值

  可以得到$Vdd_B \approx 0.0276294\ \ or\ \ 0.64099\ \ \mathrm{V}$。

  由於$0.0276294 < \mathrm{Vt}$ 不合理，故取$Vdd_B = 0.641\ \ V$，代入(1)式可得到 $Vdd_A \approx 0.664$

  $P = P_A + P_B = (20p \times 0.664^2 + 35p \times 0.641^2) \times 10M = 231.98755\ \ \mu W$

  因此

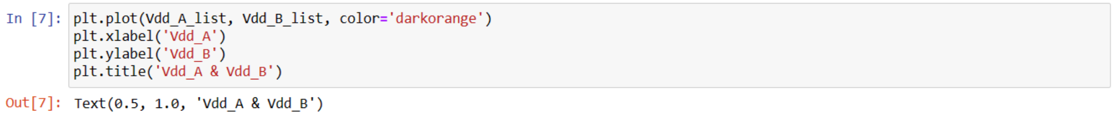
  $Vdd_A = 0.664\ \ V$

  $Vdd_B = 0.641\ \ V$

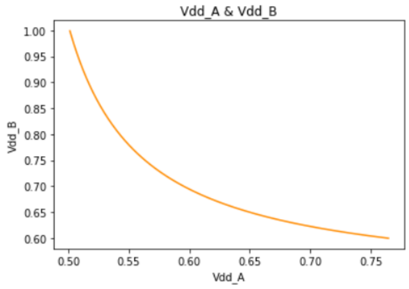
  $P_A + P_B = 231.98755\ \ \mu W$

可以發現微分等於零找極值與迭代的做法答案相近，但之間還是有些微不同。迭代法的誤差可能來自於迭代的解析度不夠，或是 epsilon 的值所造成。但迭代法的好處是運算速度較快，可以較快求解出答案。

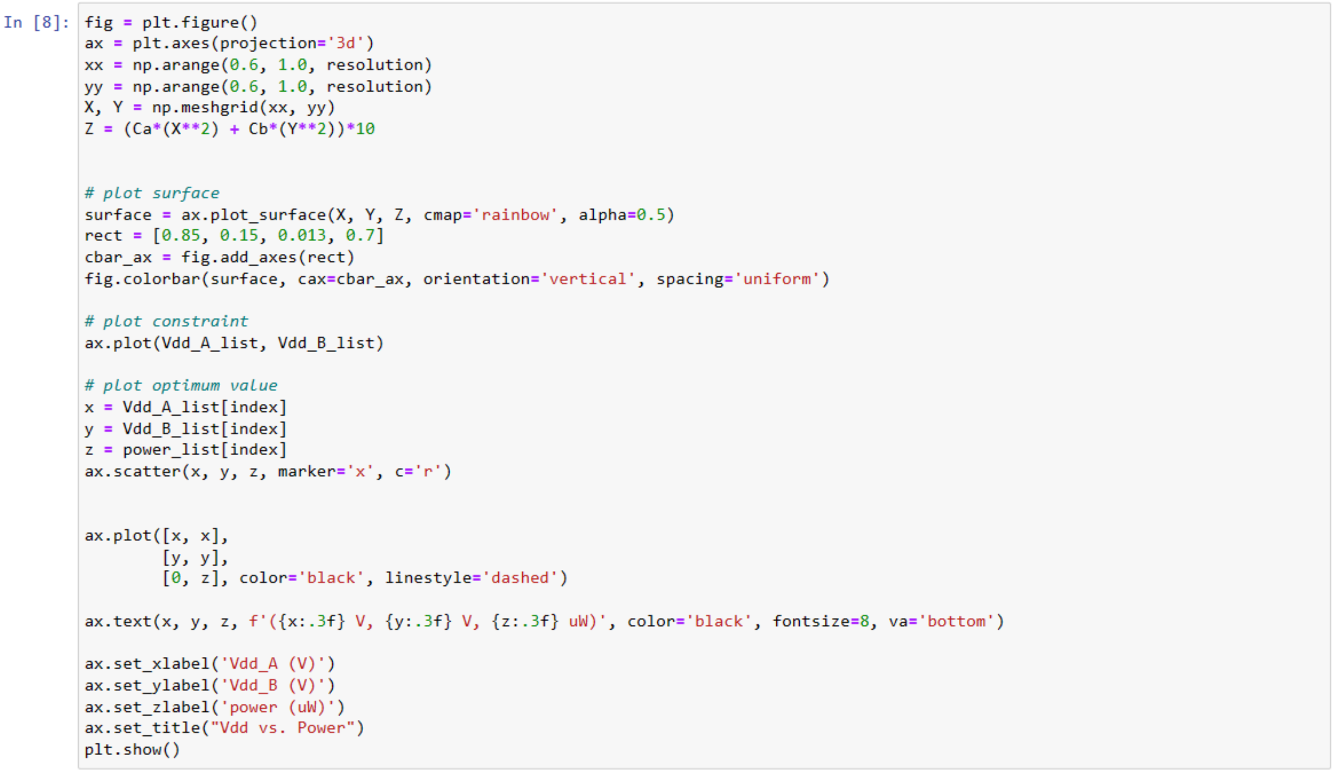
繪製Vdd_A與Vdd_B的關係圖:



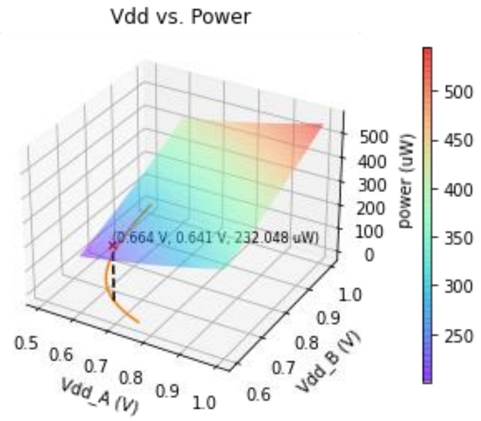
```
In [7]: plt.plot(Vdd_A_list, Vdd_B_list, color='darkorange')
        plt.xlabel('Vdd_A')
        plt.ylabel('Vdd_B')
        plt.title('Vdd_A & Vdd_B')

Out[7]: Text(0.5, 1.0, 'Vdd_A & Vdd_B')
```



繪製Vdd_A與Vdd_B代入計算後生成 Power 的平面、Vdd_A和Vdd_B的限制條件(橘線)，以及迭代後找到的 Power 最低的點(紅色×)。其中，Vdd_A和Vdd_B的限制條件(橘線)投影在 power 平面的最低點，即為所求的點，也就是$P_A + P_B$為最小的情況。



In [8]:
```
fig = plt.figure()
ax = plt.axes(projection='3d')
xx = np.arange(0.6, 1.0, resolution)
yy = np.arange(0.6, 1.0, resolution)
X, Y = np.meshgrid(xx, yy)
Z = (Ca*(X**2) + Cb*(Y**2))*10


# plot surface
surface = ax.plot_surface(X, Y, Z, cmap='rainbow', alpha=0.5)
rect = [0.85, 0.15, 0.013, 0.7]
cbar_ax = fig.add_axes(rect)
fig.colorbar(surface, cax=cbar_ax, orientation='vertical', spacing='uniform')

# plot constraint
ax.plot(Vdd_A_list, Vdd_B_list)

# plot optimum value
x = Vdd_A_list[index]
y = Vdd_B_list[index]
z = power_list[index]
ax.scatter(x, y, z, marker='x', c='r')


ax.plot([x, x],
        [y, y],
        [0, z], color='black', linestyle='dashed')

ax.text(x, y, z, f'({x:.3f} V, {y:.3f} V, {z:.3f} uW)', color='black', fontsize=8, va='bottom')

ax.set_xlabel('Vdd_A (V)')
ax.set_ylabel('Vdd_B (V)')
ax.set_zlabel('power (uW)')
ax.set_title("Vdd vs. Power")
plt.show()
```



$$Vdd_A = 0.664\ \ V$$
$$Vdd_B = 0.641\ \ V$$
$$P_A + P_B = 232.048\ \ \mu W$$

- **Calculate the power reduction ratio, P_2/P_0**

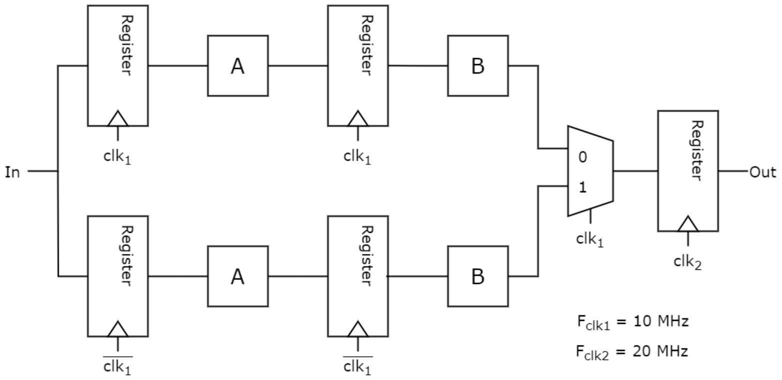
假設 Register 與 2-to-1 multiplexer 的 Vdd 為 1.0 V。

$$P_2 = 2 \times P_{reg(clk1)} + P_{mux} + 2 \times (P_A + P_B) + P_{reg(clk2)}$$
$$= [2 \times C_{reg(clk1)} \times Vdd_{reg}{}^2 + C_{mux} \times Vdd_{mux}{}^2 + 2 \times (C_A \times Vdd_A{}^2 + C_B \times Vdd_B{}^2)] \times F + C_{reg(clk2)} \times Vdd_{reg}{}^2 \times 2F$$
$$= [2 \times 0.05p \times 20 \times 1.0^2 + 0.05p \times 20 \times 1.0^2 + 2 \times (20p \times 0.664^2 + 35p \times 0.641^2)] \times 10M + 0.05p \times 20 \times 1.0^2 \times 20M = 513.9751\ \ \mu W \approx 513.98\ \ \mu W$$

$$\therefore \text{Power reduction ratio} = \frac{P_2}{P_0} = \frac{513.98\ \ \mu W}{1140\ \ \mu W} \approx 45.1\%$$

(c) Try to combine (a) and (b) to design a parallel-pipeline version while maintaining the data rate(p3). Show the block diagram, explain the operation, and calculate the power reduction ratio, P3/P0 (40%)

- **Block diagram**



結合 pipeline 和 parallel 的概念，把 b 小題架構中的 A, B 模組之間加一個 pipeline register，使得 A, B 模組運算的時間可以拉得更長 Vdd 就可以在降低，power dissipation 也可以跟著降低。

Delay time of A $\le 2\times 50 - 0.4 = 99.6\ \ \text{ns}$

$$99.6n = \frac{K_A}{Vdd_A - V_t} = \frac{14\text{n}}{Vdd_A - 0.3} \rightarrow Vdd_A \approx 0.44\ \ V$$

Delay time of B $\le 2\times 50 - (0.4 + 0.4) = 99.2\ \ \text{ns}$

$$99.2n = \frac{K_B}{Vdd_B - V_t} = \frac{20.72\text{n}}{Vdd_B - 0.3} \rightarrow Vdd_B \approx 0.51\ \ V$$

根據以上算式，Vdd_A = 0.44 V，Vdd_B = 0.51 V。

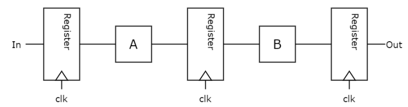
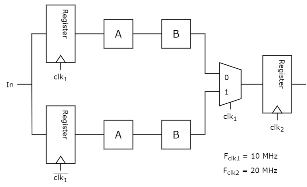
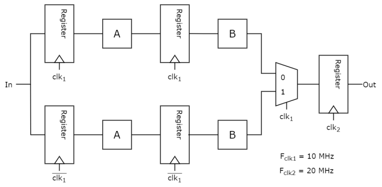
- **Calculate the power reduction ratio, P_3/P_0**

假設 Register 與 2-to-1 multiplexer 的 Vdd 為 1.0 V

$$P_3 = 4\times P_{reg(clk1)} + P_{mux} + 2\times(P_A + P_B) + P_{reg(clk2)}$$
$$= [4\times C_{reg(clk1)}\times Vdd_{reg}{}^2 + C_{mux}\times Vdd_{mux}{}^2 + 2\times\left(C_A\times Vdd_A{}^2 + C_B\times Vdd_B{}^2\right)]\times F + C_{reg(clk2)}\times Vdd_{reg}{}^2\times 2F$$
$$= [4\times 0.05p\times 20\times 1.0^2 + 0.05p\times 20\times 1.0^2 + 2\times(20p\times 0.44^2 + 35p\times 0.51^2)]\times 10M + 0.05p\times 20\times 1.0^2\times 20M = 329.51\ \ \mu W$$

$$\therefore \text{Power reduction ratio} = \frac{P_3}{P_0} = \frac{329.51\ \ \mu W}{1140\ \ \mu W} \approx 28.9\%$$

Conclusion

	(a) Pipeline	**(b) Parallel**	**(c) Pipeline + Parallel**
Block Diagram			
Vdd$_A$	0.58 V	0.66 V	0.44 V
Vdd$_B$	0.72 V	0.64 V	0.51 V
Frequency	F$_{clk}$ = 20M Hz	F$_{clk1}$=10 MHz/F$_{clk2}$=20 MHz	F$_{clk1}$=10M Hz/F$_{clk2}$=20 MHz
Power ratio	$P_1/P_0 \approx\ 48.9\%$	$P_2/P_0 \approx\ 45.1\%$	$P_3/P_0 \approx\ 28.9\%$
Area	$\text{Area}_A + \text{Area}_B$ $+60\times\text{Area}_{reg}$	$2\times(\text{Area}_A + \text{Area}_B)$ $+20\times\text{Area}_{mux}$ $+60\times\text{Area}_{reg}$	$2(\text{Area}_A + \text{Area}_B)$ $+20\times\text{Area}_{mux}$ $+100\times\text{Area}_{reg}$