姓名:劉世棠　　學號:R07943095　　系級:EDA 碩一

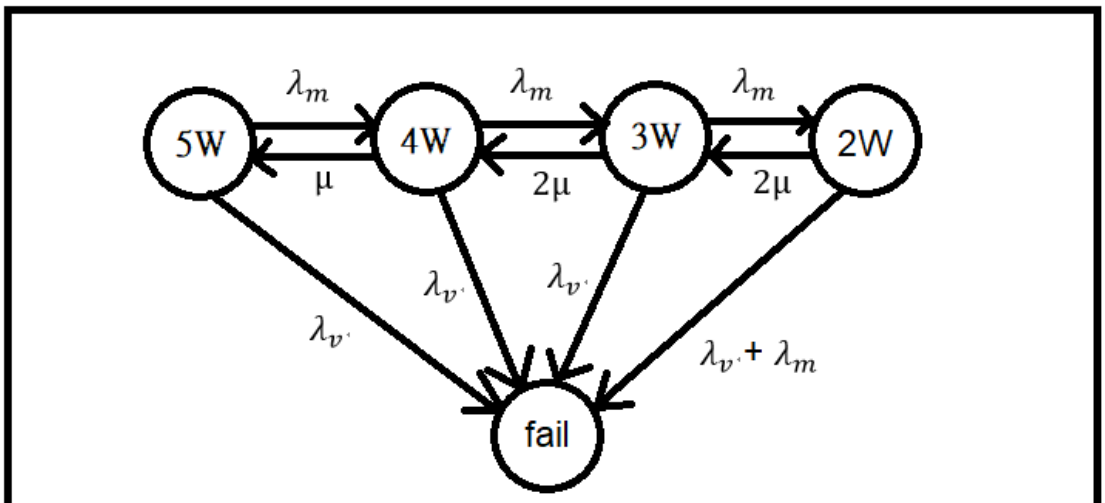
1. 第一題:

利用題目附上的公式可以求得 upper bound(原因為假設每一條路徑都不通)，而所有路徑有 ABDEF、ABDI、ACDEF、ACDI、AGF、HDEF、HDI 等七條路徑，元件 X 的可靠度將以R_X表示，故答案為:

$$R_{sys} \leq 1-(1-R_A R_B R_D R_E R_F)\times(1-R_A R_B R_D R_I)\times(1-R_A R_C R_D R_E R_F)\times(1-R_A R_C R_D R_I)\times(1-R_A R_G R_F)\times(1-R_H R_D R_E R_F)\times(1-R_H R_D R_I)$$

2. 第二題:

這邊假設 system fail 就無法回復，原因為 system fail 可能無法得知，甚至整個系統直接損毀(意外過載):



假設系統壞掉就無法復原(因為題目未說明另外的偵測機制)

3. 第三題:

(i) 為了要能使系統運作，所以並行的 module 不能全部都損壞，至少要有一個 module 是好的(維持系統運作)，故答案為 4 個 faulty modules。

(ii) 假設每個 module 都有自己的 fault detection 機制，故要求最多不能全部壞掉，故答案為 4 個 faulty modules。

(iii) 為了要能偵測錯誤，所以至少要兩個正常在運作的模組中必須要有一個是對的，且為了修復之後還能正確運作，最多也只能有四個是錯的，故答案為在不考慮運行的兩個 module 同時損壞的情況下最多可以有 4 個 faulty modules。

4. 第四題:

(i) 假設這題的 reliability 是用$e^{-(failure\ rate)\ t}$去估計的，且假設出廠後為 failure rate 是個常數(至少在三年內)，則第一年約有$\left(1-(0.73)^{\frac{1}{3}}\right)\times 100\%$需要回廠修復。

(ii) 假設這題的 reliability 是用$e^{-(failure\ rate)\ t}$去估計的，且假設出廠後為 failure rate 是個常數(至少在三年內)，則 MTTF = 1/(failure rate) = -t/ln(0.73)且

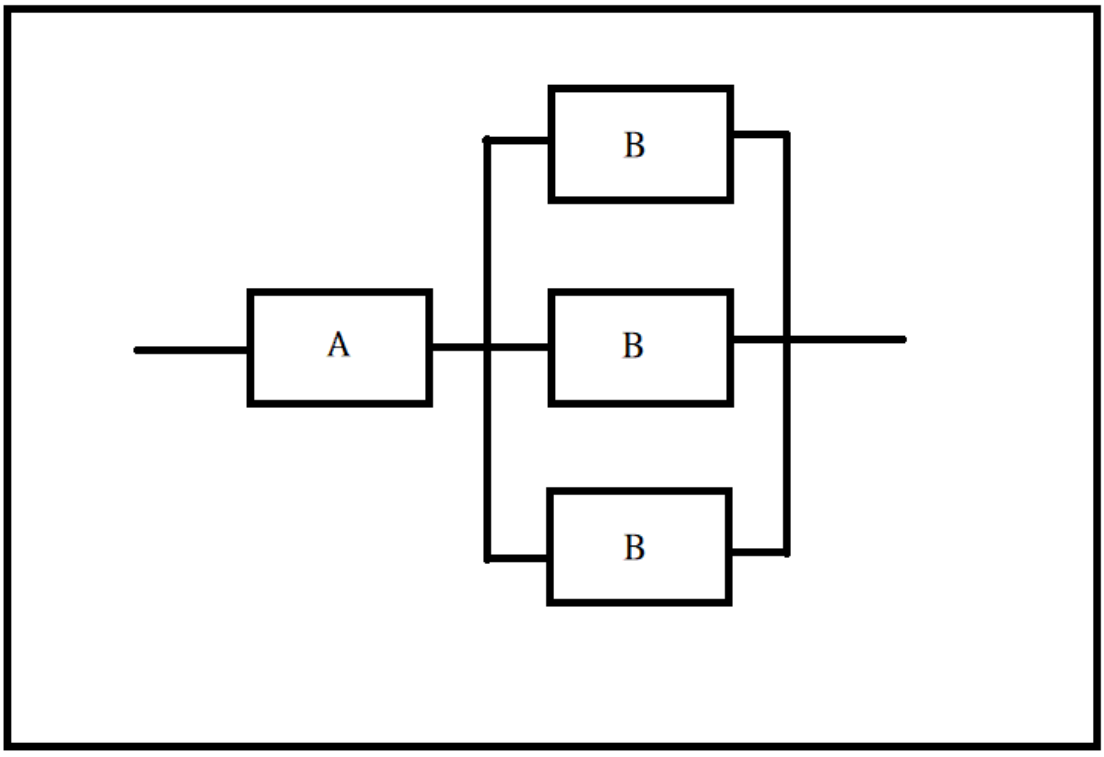
t=3*365，則 MTTF 約為 3479.385 天。

(iii) MTBF= MTTF+MTTR= 2－(3*365)/ln(0.73)= 3481.385 天

(iv) 因為題目未假設清楚，所以我便自己做假設，如果說系統要兩個平行的機器都壞掉才算損壞，則 failure prob 為$(1 - (0.73)^{1/3})^2 \times 100\%$，若只要單一壞掉就算壞掉則為$\left(1 - \left((0.73)^{\frac{1}{3}}\right)^2 \right) \times 100\%$ ，以上皆假設無法修復。

5. 第五題:

這題看起來瓶頸是在 B，所以我將 B 增加為三個平行接在一起，此時的可靠度從0.99×0.85變成$0.99 \times (1 - (1 - 0.85)^3)$，圖示如下:

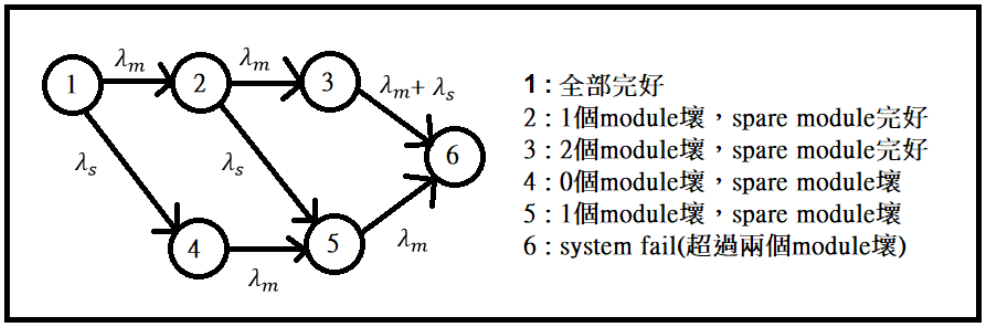


6. 第六題(**以下皆假設 R(t)=$e^{-(failure\ rate)t}$，且共三個 module 加上一個額外的 spare**):

(1) 做 reliability 分析:

因為系統在「4 個 module 正常運作」或「3 個 module 正常運作」或「2 個 module 正常運作」時會正常運作:

$$R(t) = e^{-3\lambda_m t}e^{-\lambda_s t} + \left(3e^{-2\lambda_m t}\left(1-e^{-\lambda_m t}\right)e^{-\lambda_s t} + e^{-3\lambda_m t}\left(1-e^{-\lambda_s t}\right)\right) + \left(3e^{-\lambda_m t}\left(1-e^{-\lambda_m t}\right)^2 e^{-\lambda_s t} + 3e^{-2\lambda_m t}\left(1-e^{-\lambda_m t}\right)\left(1-e^{-\lambda_s t}\right)\right)$$



(2) 做 availability 分析:

因為系統失敗時會自動停止，即便有 repair 的機制，但是系統關閉便無法回復，而在三個 module 壞掉且 voter 完好或 voter 壞掉時即失敗，故 availability 為:

在 system 正常運作時，各個元件的 availability 可以用以下去估算:

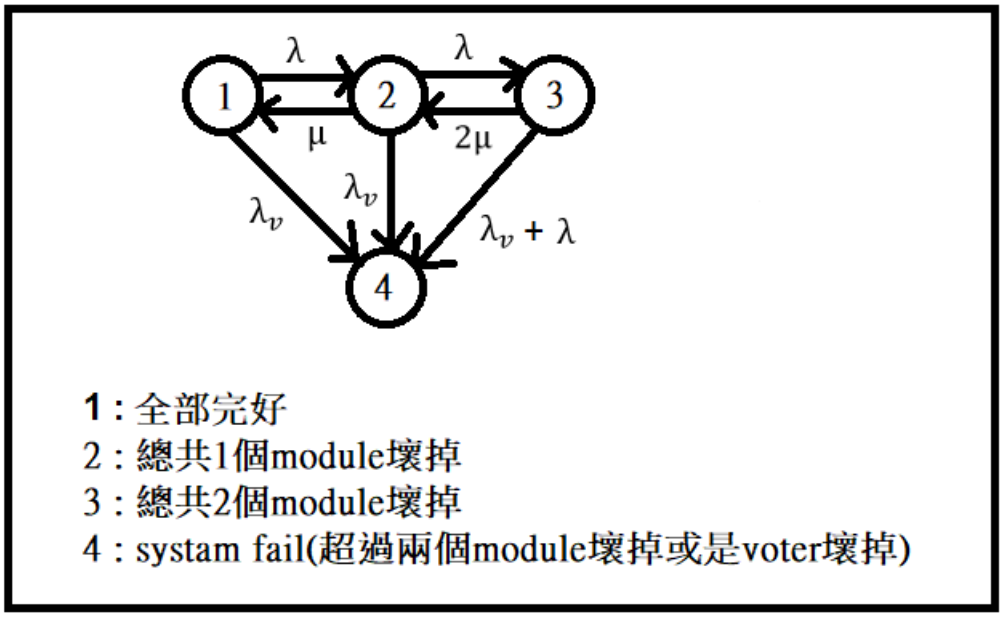
$$\begin{cases} A(t) = A_1(t) = A_2(t) = A_3(t) = A_{spare}(t) = \dfrac{\mu}{\lambda+\mu} \\ A_v(t) = \dfrac{\mu}{\lambda_v+\mu} \end{cases}$$

故λ_{sys}為:

$$\lambda_{sys} = 1 - \big(1 - A_v(t)\big) - A_v(t)\big(1 - A(t)\big)^3$$

$$= 1 - \frac{\lambda_v}{\lambda_v+\mu} - \frac{\mu}{\lambda_v+\mu} \times \left(\frac{\lambda}{\lambda+\mu}\right)^3$$

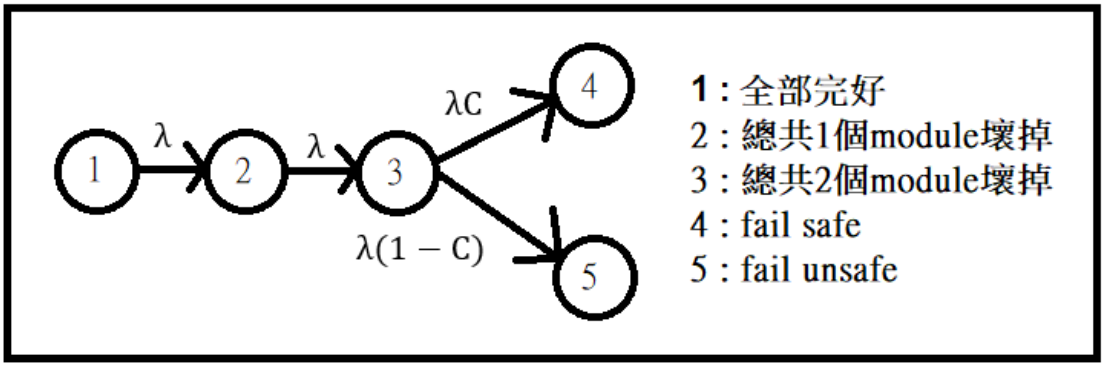
故答案為 : $\mathbf{A_{sys}(t) = e^{-\lambda_{sys}t}}$



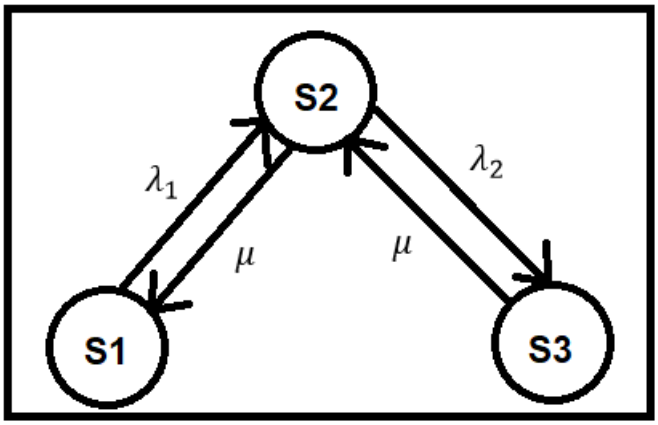
(3) 做 safety 分析:

只有當三個 module 壞掉且 disagreement detector 失敗時才會 fail unsafe，所以 safety 為:

$$R(t) = 1 - (1 - C) \times \left(1 - e^{-\lambda t}\right)^3$$



7. 第七題:



8. 第八題:

否，因為 self dual 要滿足 f(X)=f’(X’)，而

f’(a’,b’,c’)=a+b+c，故並非 self dual function。

9. 第九題(單個的 reliability $R(t) = e^{-\lambda t}$,$\lambda = failure\ rate$)，不過因為底下 hint 是$R(t) = e^{-\lambda}$，與λ是個常數矛盾，故修改為 $e^{-\lambda t}$:

(1) 因為只要有一個完好就可以順利運行，故:

$$R_{sys}(t) = 1 - (1 - e^{-\lambda t})^3$$

(2) 因為$R_{sys}(t) = e^{-\lambda_{sys} t} = 1 - (1 - e^{-\lambda t})^3$ ，故

$$\lambda_{sys} = -\frac{\ln\left(R_{sys}(t)\right)}{t} = -\frac{\ln\left(1 - (1 - e^{-\lambda t})^3\right)}{t}$$

(3) 如果說三個的 failure rate 分別為λ_1 λ_2 λ_3，則

$R_{sys}(t) = 1 - (1 - e^{-\lambda_1 t})(1 - e^{-\lambda_2 t})(1 - e^{-\lambda_3 t})$ ，

故:$\lambda_{sys} = -\frac{\ln\left(R_{sys}(t)\right)}{t} = -\frac{\ln\left(1-(1-e^{-\lambda_1 t})(1-e^{-\lambda_2 t})(1-e^{-\lambda_3 t})\right)}{t}$

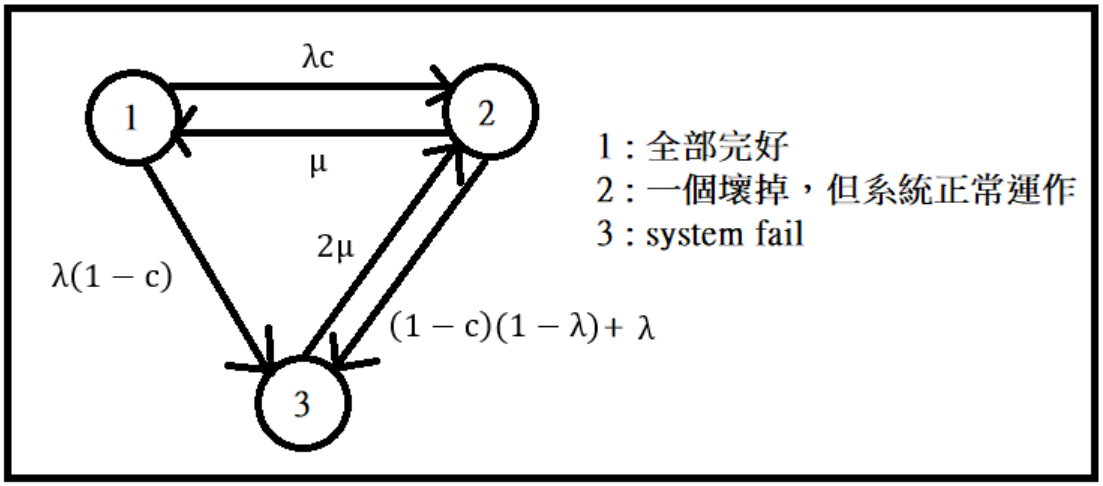
10. 第十題:

（1） 這邊因為假設 comparator C 跟 voter 都是完美的，整體的 reliability 有三種可能，全部正常(係數為 1)、一個不正常(共四種，係數為 4)、兩個不正常的情況(兩個不正常共有六組合，不過 module2 和 module3 同時損壞不能偵測，係數為 5)，故答案為:$R_{sys} = R^4 + 4(1-R)R^3 + 5(1-R)^2R^2$

（2） 這邊與上面相比多加了一個 fault coverage 進 perfect comparator 之中，整體的 reliability 有三種可能，全部正常(係數依然為 1)、一個不正常(共四種且不管有沒有正確偵測一定會有兩個正確的數值餵進 voter，係數依然為 4)、兩個不正常的情況(兩個不正常共有六組合，不過 module2 和 module3 同時損壞不能偵測，當 module1 和 a 有一個壞掉時要乘上 c 且共有四種，最後 module1 和 a 都壞掉時無關，係數變為 0+4c+1)，答案為: $R_{sys} = R^4 + 4(1-R)R^3 + (1+4c)(1-R)^2R^2$

11. 第十一題:

(1) 畫出 Markov model:



(2) 將第一題的 Markov chain 化為矩陣:

$$M = \begin{bmatrix} -\lambda & 2\lambda & 0 \\ c\lambda & -1+c-c\lambda-2\lambda & 4\lambda \\ \lambda(1-c) & (1-c)(1-\lambda)+\lambda & -4\lambda \end{bmatrix}$$

$$\begin{cases} -\lambda P_1(\infty) + 2\lambda P_2(\infty) = 0 \\ c\lambda P_1(\infty) + (-1+c-c\lambda-2\lambda)P_2(\infty) + 4\lambda P_3(\infty) = 0 \\ (1-c)\lambda P_1(\infty) + \big((1-c)(1-\lambda)+\lambda\big)P_2(\infty) - 4\lambda P_3(\infty) = 0 \end{cases}$$

配合機率為守恆的概念可得到:

$$\begin{cases} P_1(\infty) = 2P_2(\infty) \\ P_2(\infty) = \frac{4\lambda}{1-c-c\lambda+2\lambda} P_3(\infty) \\ P_1(\infty) + P_2(\infty) + P_3(\infty) = 1 \end{cases}$$

可得到 ： $P_3(\infty) = \frac{1}{\left(1+3\times\frac{4\lambda}{1-c-c\lambda+2\lambda}\right)} = \frac{1-c-c\lambda+2\lambda}{1-c-c\lambda+14\lambda}$

因為除了$P_3(\infty)$之外都是系統輸出正常，故:

$$\mathrm{A}(\infty) = 1 - P_3(\infty) = 1 - \frac{1-c-c\lambda+2\lambda}{1-c-c\lambda+14\lambda}$$

12. 第十二題:

(A) 此題因為題目要我們用表格列出，且總共有三個 process，故會有三個表格，為了避免模糊，我將會先於此說明我的邏輯:

(1) SSN 因為未找到明確的更新方式，故我假設 SSN 在送出 message 時皆直接做加一的動作。

(2) 因為 SSN 與 RSN 皆代表下一個要使用的數字，所以當 t0 時 P3 的 SSN 為 2(因為此刻使用的 SSN 為 1，下次的 SSN 為 2)，這邊可能會與標準答案的定義有提前一個 time frame，故於此說明之。

(3) 假設這邊的 ack/RSN、ack 皆運作順利。

(4) 有任何變數的變動皆會用紅字標註。

(5) **因為題目有些模糊，故假設 m1 包含 message 傳送、對方回傳 ack/RSN、自己收到對方的 ack/RSN 再回傳 ack 等動作。**

P1:

t1 ： 接收來自 P3 的 message(m1)，更新 RSN 與來自 P3 最高的 SSN 與更新回傳的 RSN

t3 ： 接收來自 P2 的 message(m2)，更新 RSN 與來自 P2 最高的 SSN 與更新回傳的 RSN

t4 ： 發送給 P3　　message(m3)，更新 Log 與更新 SSN

t7 ： 發送給 P2　　message(m5)，更新 Log 與更新 SSN

t9 ： 發送給 P3　　message(m6)，更新 Log 與更新 SSN

time	SSN_1	RSN_1	Log_1(m, Pj, ssn)	T_1'(Pj, SSN_high)	T_1''(m, RSN)
t0	1	1	ϕ	ϕ	ϕ
t1	1	2	ϕ	$(P3,1)$	$(m1,1)$
t2	1	2	ϕ	$(P3,1)$	$(m1,1)$
t3	1	3	ϕ	$\begin{cases}(P3,1)\\(P2,1)\end{cases}$	$\begin{cases}(m1,1)\\(m2,2)\end{cases}$
t4	2	3	$(m3,P3,1)$	$\begin{cases}(P3,1)\\(P2,1)\end{cases}$	$\begin{cases}(m1,1)\\(m2,2)\end{cases}$
t5	2	3	$(m3,P3,1)$	$\begin{cases}(P3,1)\\(P2,1)\end{cases}$	$\begin{cases}(m1,1)\\(m2,2)\end{cases}$
t6	2	3	$(m3,P3,1)$	$\begin{cases}(P3,1)\\(P2,1)\end{cases}$	$\begin{cases}(m1,1)\\(m2,2)\end{cases}$
t7	3	3	$\begin{cases}(m3,P3,1)\\(m5,P2,2)\end{cases}$	$\begin{cases}(P3,1)\\(P2,1)\end{cases}$	$\begin{cases}(m1,1)\\(m2,2)\end{cases}$
t8	3	3	$\begin{cases}(m3,P3,1)\\(m5,P2,2)\end{cases}$	$\begin{cases}(P3,1)\\(P2,1)\end{cases}$	$\begin{cases}(m1,1)\\(m2,2)\end{cases}$
t9	4	3	$\begin{cases}(m3,P3,1)\\(m5,P2,2)\\(m6,P3,3)\end{cases}$	$\begin{cases}(P3,1)\\(P2,1)\end{cases}$	$\begin{cases}(m1,1)\\(m2,2)\end{cases}$
t10	4	3	$\begin{cases}(m3,P3,1)\\(m5,P2,2)\\(m6,P3,3)\end{cases}$	$\begin{cases}(P3,1)\\(P2,1)\end{cases}$	$\begin{cases}(m1,1)\\(m2,2)\end{cases}$

P2:

t2 ： 發送給 P1　　message(m2)，更新 Log 與更新 SSN

t6 ： 發送給 P3　　message(m4)，更新 Log 與更新 SSN

t8 ： 接收來自 P1 的 message(m5)，更新 RSN 與來自 P1 最高的 SSN 與更新回傳的 RSN

time	SSN_2	RSN_2	Log_2(m,Pj,ssn)	T_2'(Pj,SSN_high)	T_2''(m,RSN)
t0	1	1	ϕ	ϕ	ϕ
t1	1	1	ϕ	ϕ	ϕ
t2	2	1	$(m2,P1,1)$	ϕ	ϕ
t3	2	1	$(m2,P1,1)$	ϕ	ϕ
t4	2	1	$(m2,P1,1)$	ϕ	ϕ
t5	2	1	$(m2,P1,1)$	ϕ	ϕ
t6	3	1	$\begin{cases}(m2,P1,1)\\(m4,P3,2)\end{cases}$	ϕ	ϕ
t7	3	1	$\begin{cases}(m2,P1,1)\\(m4,P3,2)\end{cases}$	ϕ	ϕ
t8	3	2	$\begin{cases}(m2,P1,1)\\(m4,P3,2)\end{cases}$	(P1,2)	(m5,1)
t9	3	2	$\begin{cases}(m2,P1,1)\\(m4,P3,2)\end{cases}$	(P1,2)	(m5,1)
t10	3	2	$\begin{cases}(m2,P1,1)\\(m4,P3,2)\end{cases}$	(P1,2)	(m5,1)

P2:

t1 ： 發送給 P1　　message(m1)，更新 Log 與更新 SSN

t5 ： 接收來自 P1 的 message(m3)，更新 RSN 與來自 P1 最高的 SSN 與更新回傳的 RSN

t7 ： 接收來自 P2 的 message(m4)，更新 RSN 與來自 P2 最高的 SSN 與更新回傳的 RSN

t10: 接收來自 P1 的 message(m6)，更新 RSN 與來自 P1 最高的 SSN 與更新回傳的 RSN

time	SSN_3	RSN_3	Log_3(m,Pj,ssn)	T_3'(Pj,SSN_high)	T_3''(m,RSN)
t0	2	1	$(m1,P1,1)$	ϕ	ϕ
t1	2	1	$(m1,P1,1)$	ϕ	ϕ
t2	2	1	$(m1,P1,1)$	ϕ	ϕ
t3	2	1	$(m1,P1,1)$	ϕ	ϕ
t4	2	1	$(m1,P1,1)$	ϕ	ϕ
t5	2	2	$(m1,P1,1)$	(P1,1)	(m3,1)
t6	2	2	$(m1,P1,1)$	(P1,1)	(m3,1)
t7	2	3	$(m1,P1,1)$	(P1,1) $(P2,2)$	(m3,1) (m4,2)
t8	2	3	$(m1,P1,1)$	(P1,1) $(P2,2)$	(m3,1) (m4,2)
t9	2	3	$(m1,P1,1)$	(P1,1) $(P2,2)$	(m3,1) (m4,2)
t10	2	4	$(m1,P1,1)$	(P1,3) $(P2,2)$	(m3,1) (m4,2) (m6,3)

(B) 說明 P3 收到 m6 後立刻 crash(尚未傳出 ack/RSN 訊號)要如何 recovery:

因為 P3 的 check point 是在 t0，故會先回到 t0 的狀態，此時 P3 有存 message 和 RSN 在 local，故會將有完整的 message log 與 RSN 的資料重新傳送，並將廣播去要求分散在其他 process 的 log 以回復狀態，當其他 process 收到 message 會比對 SSN，如果 SSN 低於紀錄的最高 SSN，這個 process 會複製訊息送出，最後完成復原。