

# 2022 IC Design Contest

## Cell-Based IC Design Category for Undergraduate Level

### *Job Assignment Machine*

#### 1.問題描述

派工機(Job Assignment Machine) (後文以 **JAM 表示**) 的應用相當廣泛，當有  $n$  件工作需要完成，而  $n$  個工人對每件工作可能有不同的工作成本，如何來指派那一個工人去執行那一件工作，以期達到最低成本，此即派工機的目的。

解決派工問題最直覺的方法是計算所有可能組合的成本，然後可找到最低成本的組合。本題題目輸入工人工作成本資訊，請以窮舉法列出所有的配對組合，然後找到最低的成本以及符合最低成本的組合數量。

有關 JAM 詳細規格將描述於後。表一為本電路各輸入、輸出信號的功能說明。各參賽隊伍必須依照下一節所指定的設計規格完成設計驗證。

本次 IC 設計競賽比賽時間為上午 08:30 到下午 20:30。當 IC 設計競賽結束後，本題會根據第三節中的評分標準進行評分。為了評分作業的方便，各參賽隊伍應參考附錄 C 中所列的要求，附上評分所需要的檔案。











					
	12	22	34	54	12
	45	21	97	98	34
	54	88	21	22	34
	12	43	57	21	33
	35	98	32	1	13

圖 1、工作成本表格

## 2.設計規格

### 2.1 系統方塊圖

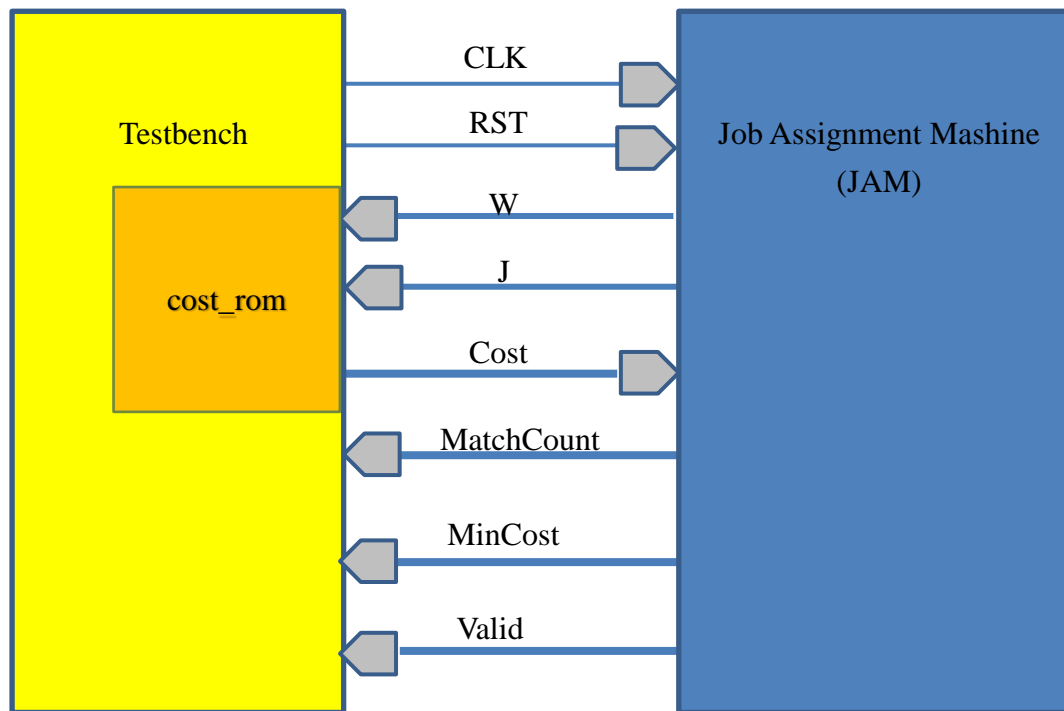


圖 2、Job Assignment Machine 之系統方塊圖

### 2.2 輸入/輸出介面

表一、輸入/輸出訊號

Signal Name	I/O	Width	Simple Description
CLK	I	1	Clock Signal (positive edge trigger)
RST	I	1	Reset Signal (active high)。由 testbench 提供，拉高 2 cycle 後恢復為 low。
W	O	3	指定取得第 W 位工人成本資料， $0 \leq W \leq 7$
J	O	3	指定取得第 J 項成本資料， $0 \leq J \leq 7$
Cost	I	7	成本數值，當 W 及 J 設定完成，Cost 回應第 W 位工人在第 J 項工作的成本數值。Cost 為無號二進位整數，數值範圍為 0 到 100 的正整數。
MatchCount	O	4	輸出符合最小成本的可能組合的數量。
MinCost	O	10	輸出最小總工作成本的數值。 MinCost 為無號二進位整數 testbench 的最小總工作成本不會超過 1024。
Valid	O	1	當 Valid 為 high，表示目前 MatchCount 以及 MinCost 為有效的輸出，testbench 會在下一 cycle 立刻結束模擬。

## 2.3 系統描述

以窮舉方式產生所有的配對組合，計算符合最小工作成本的組合數量，**本題需針對  $n=8$  (指派 8 個工人去完成 8 件工作) 的案例進行派工機 JAM 電路作設計。**

### 2.3.1 JAM 電路原始成本資料的輸入

本題工作成本資料儲存在一塊  $8 \times 8$  的**非同步** cost\_rom 中，系統於重置後，JAM 電路指定 W 及 J 訊號取得第 W 位工人在第 J 項工作的成本資料。W 及 J 數值範圍皆為 0 到 7。

成本資料從 Cost 輸入，直接回應 W 及 J 的內容，不受 CLK 限制。

JAM 電路可重覆取用 cost\_rom 內資料。



圖 3、Job Assignment Machine 之輸入訊號波形

### 2.3.2 JAM 電路計算結果的輸出

JAM 計算出最小總工作成本，以及符合此成本的組合數量後，分別從 MinCost 及 MatchCount 輸出，同時拉高 Valid 訊號，testbench 收到 Valid 訊號即會停止模擬並比對結果正確性。

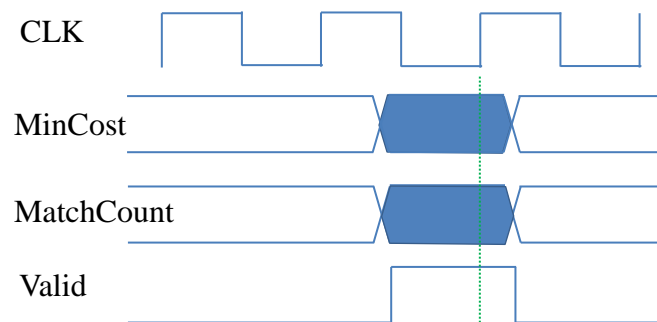


圖 4、Job Assignment Machine 之輸出訊號波形

### 2.3.3 全排列演算法

給定  $n$  個數字，最多可以有  $n!$  種排列，列出這  $n$  個數字所有的排列組合，稱為全排列。以  $n=4$  為例，**[0,1,2,3]** 的全排列為：

**[0,1,2,3] [0,1,3,2] [0,2,1,3] [0,2,3,1] [0,3,1,2] [0,3,2,1]**  
**[1,0,2,3] [1,0,3,2] [1,2,0,3] [1,2,3,0] [1,3,0,2] [1,3,2,0]**  
**[2,0,1,3] [2,0,3,1] [2,1,0,3] [2,1,3,0] [2,3,0,1] [2,3,1,0]**  
**[3,0,1,2] [3,0,2,1] [3,1,0,2] [3,1,2,0] [3,2,0,1] [3,2,1,0]**

產生全排列的方式有很多種，在此介紹方法為字典序演算法，字典序的意思是，將序列元素依數字大小排列，比如說**[2,3,1,0]**就排在**[2,3,0,1]**後面，如果有一個方法，可以找到任一序列的下一序列，就可以依續找到所有的序列，這就是字典序演算法。

字典序演算法分成 3 步驟，以下範例以  $n=7$ ，七個數字**[0, 1, 2, 3, 4, 5, 6]**為例，假設目前序列為 **[3, 0, 4, 6, 5, 2, 1]**，求下一字典序列。

1. 從右邊開始，找到第一組相鄰且右邊比左邊大的位置：

以上面序列為例，**[2, 1]** 右邊較小不合條件；

**[5, 2]** 不合條件；

**[6, 5]** 不合條件；

**[4, 6]** 右邊較大，符合條件

我們稱 **4** 的位置為**替換點**，且 **4** 為替換數。

**[3, 0, 4, 6, 5, 2, 1]**

2. 在**替換點**右邊的的數字中，找到比替換數大的最小數字，將之和替換數交換

上面例子，替換數為 **4**，右邊的數字為**[6, 5, 2, 1]**，這幾個數中比 **4** 大的最小數為 **5**

把 **4** 和 **5** 交換，序列變成**[3, 0, 5, 6, 4, 2, 1]**

3. 最後把**替換點**後的數字前後順序翻轉過來，即可得下一字典序列。

把**[6, 4, 2, 1]**翻轉過來，序列就變成**[3, 0, 5, 1, 2, 4, 6]**，此序列就是原序列**[3, 0, 4, 6, 5, 2, 1]**的下一序列。

### 2.3.4 計算總工作成本

當計算出某一序列後，即可依此序列數值分配工作給每一位工人，總工作成本計算方式是將每一位工人分配到的工作的成本相加。

比如  $n=5$  序列**[3, 2, 4, 0, 1]**，依下圖表格可計算此組合之成本為  $54+97+34+12+98$

	12	22	34	54	12
	45	21	97	98	34
	54	88	21	22	34
	12	43	57	21	33
	35	98	32	1	13

### 3.評分標準

設計目標有二項，

目標一、總模擬 cycle 數小於 600000 cycle

模擬結束會回報總模擬 cycle 數，總模擬 cycle 數取 3 組 pattern 模擬最長那組之 cycle 數

```
-----
Get Valid at cycle: 357751
receive MinCost/MatchCount= 119/ 3 , golden MinCost/MatchCount= 119/ 3
-----
*****
**  FUNCTION  CORRECT  **
*****
```

目標二、合成後面積小於 10000um<sup>2</sup>

Design compile report area 範例: dc\_shell> report\_area

Combinational area:	5533.524058
Buf/Inv area:	612.761391
Noncombinational area:	3518.710207
Macro/Black Box area:	0.000000
Net Interconnect area:	90300.458740
Total cell area:	9052.234265
Total area:	99352.693005

評分方式會依設計完成程度，分成 A、B、C、D 四種等級，排名順序為 A>B>C>D。

本題設定 clock 週期時間為 10ns，參賽者不可調整 clock 週期時間。

✧ **等級 A：** 等級 A 條件：

- a、在 clock 週期為 10ns 環境下，Gate-Level 與 RTL 模擬完全正確。
- b、滿足設計目標一，總模擬 cycle 數小於 600000 cycle。
- c、滿足設計目標二，合成後面積小於 10000um<sup>2</sup>

等級 A 之評分方法：

依繳交時間由早到晚排序。

✧ **等級 B：** 等級 B 條件：

- a、在 clock 週期為 10ns 環境下，Gate-Level 與 RTL 模擬完全正確。
- b、滿足設計目標一，總模擬 cycle 數小於 600000 cycle。
- c、合成後面積大於 10000um<sup>2</sup>

等級 B 之評分方法：

依合成面積由小到大排序。

✧ **等級 C：** 等級 C 條件：

- a、在 clock 週期為 10ns 的條件下，Gate-Level 與 RTL 模擬完全正確。
- b、總模擬 cycle 數大於 600000 cycle。

等級 C 之評分方法：

依(合成面積 x cycle)由小到大排序。

◇ **等級 D：** 等級 D 條件：

a、 RTL 模擬正確，但 Gate-Level 模擬錯誤。

**等級 D 之評分方法：**

依 cycle 由小到大排序。

## 附錄 A、設計檔說明

1. 下表為主辦單位所提供各參賽者的設計檔

表二、設計檔案說明

檔名	說明
JAM.v	參賽者所使用的設計檔，已包含系統輸/出入埠宣告。
tb.sv	Test Bench 檔案。
cost_rom	Test Pattern 資料
.synopsys_dc.setup synopsys_dc.setup	使用 Design Compiler 做合成之初始化設定檔。參賽者請依 Library 實際擺放位置，自行修改 Search Path 的設定。注意：合成時請使用 worst case library。
JAM.sdc	Design Compiler 作合成之 Constraint 檔案。
report.000	report 檔格式，見附錄 C。
dc_syn.tcl	dc 合成參考指令
ncvlog.f	ncverilog 模擬參考參數檔案。
vcs.cmd	vcs 模擬參考指令

2. 請使用 JAM.v，進行本題電路之設計。其 Verilog 模組名稱、輸出/入埠宣告如下所示：

若有需要，參賽者可自行移除 output 的 reg 宣告。

```

module JAM (
input  CLK,
input  RST,
output reg [2:0] W,
output reg [2:0] J,
input  [6:0] Cost,
output reg [3:0] MatchCount,
output reg [9:0] MinCost,
output reg Valid );

endmodule

```

3. 本題所提供之 testbench 檔，有增加數行 define 描述如下

```
`define sdf_file " ./JAM_syn.sdf "  
`ifdef SDF  
    initial $sdf_annotate(`sdf_file , u_JAM);  
`endif
```

3.1 SDF 檔之檔名，請自行依 SDF 實際檔名及路徑進行修改後再模擬。

3.2 在 testbench 中的`ifdef SDF描述，可讓該 testbench 同時適用於RTL模擬及合成後 gate-level 模擬。參賽者在進行 gate-level 模擬時，需於模擬命令上多加一個 +define+SDF 參數方可順利模擬，範例如下

```
ncverilog tb.v JAM_syn.v -v tsmc13_neg.v +define+P1 +define+SDF
```

4. 主辦單位提供三組測試樣本供參賽者驗證設計的正確性

請自行使用+define+P1、+define+P2、+define+P3 參數作切換。

5. 請勿針對這三組測試樣本的內容作設計，比如設計中判斷 pattern 為某固定數值，或是判斷第 n 個 pattern 直接設定輸出結果等。如經發現一律不予評分。

6. 題目內所提供字典序演算法，非產生全排列唯一方法，只要能完成功能，不限定一定要使用題目的方法。

7. RTL 及 Gate-level 模擬的指令範例說明如下

RTL Simulation 時使用指令如下：

使用 P1 測試樣本之 RTL simulation 之參考指令

➤ 使用 ncverilog 模擬指令範例如下：

```
ncverilog tb.v JAM.v +define+P1
```

➤ 使用 modelsim 模擬，則是在 compile verilog 時，使用下面指令：

```
vlog tb.v
```

Gate-level simulation 參考指令如下

➤ 使用 ncverilog 模擬指令範例如下：

```
ncverilog tb.v JAM_syn.v -v tsmc13_neg.v +define+P1 +define+SDF
```

➤ NC-Verilog 使用者，若要輸出 FSDB 檔，可自行在參數中加入

```
+define+FSDB +access+r
```

modelsim 使用者，請直接使用內建波形來進行除錯。

## 附錄 B、Test Pattern

以下列出 3 組 test pattern 內容：

Pattern1：

	J0	J1	J2	J3	J4	J5	J6	J7	MinCost=119 , MatchCount=3
W0	11	25	53	41	59	32	25	59	Min cost at job serial : (in order from W0 to W7)
W1	4	11	25	11	59	31	53	11	1 7 5 0 4 3 6 2
W2	11	59	15	11	15	15	53	53	6 7 5 0 4 3 1 2
W3	4	59	32	34	53	41	34	59	6 7 5 0 4 3 2 1
W4	15	32	41	34	4	59	34	32	
W5	41	59	59	4	4	41	34	34	
W6	53	31	25	41	59	32	31	53	
W7	11	31	25	11	34	34	53	32	

Pattern2：

	J0	J1	J2	J3	J4	J5	J6	J7	MinCost=250, MatchCount=6
W0	54	59	59	59	32	40	62	40	Min cost at job serial : (in order from W0 to W7)
W1	54	32	32	79	32	38	32	62	4 1 2 0 3 5 6 7
W2	54	54	30	38	32	38	59	54	4 1 2 0 3 6 5 7
W3	30	59	32	32	62	40	45	79	4 1 2 0 7 5 6 3
W4	32	32	38	32	62	38	62	32	4 1 2 0 7 6 5 3
W5	79	45	32	62	32	32	32	59	4 1 2 3 7 5 6 0
W6	32	38	32	59	54	30	30	45	4 1 2 3 7 6 5 0
W7	30	79	32	32	62	30	45	32	

Pattern3：

	J0	J1	J2	J3	J4	J5	J6	J7	MinCost=485, MatchCount=9
W0	81	60	60	65	96	60	65	96	Min cost at job serial : (in order from W0 to W7)
W1	96	60	66	96	60	60	60	81	1 5 2 7 4 0 3 6
W2	96	66	60	99	60	81	65	65	1 5 4 7 6 0 3 2
W3	66	96	80	99	81	81	96	60	2 5 4 7 6 0 3 1
W4	81	96	65	96	60	96	60	81	3 5 2 7 4 0 1 6
W5	60	96	80	96	80	60	81	60	3 5 4 7 6 0 1 2
W6	99	60	99	65	80	80	81	66	5 1 2 7 4 0 3 6
W7	65	60	60	99	99	80	60	96	5 1 4 7 6 0 3 2
									5 4 2 7 6 0 3 1
									5 6 2 7 4 0 3 1



## 附錄 C、評分用檔案

評分所須檔案可以下幾個部份：

- (1) RTL design，即各參賽隊伍對該次競賽設計的 RTL code，若設計採模組化而有多個設計檔，請務必將合成所要用到的各 module 檔放進來，以免評審進行評分時，無法進行模擬；
- (2) Gate-Level design，即由合成軟體所產生的 gate-level netlist，以及對應的 SDF 檔；
- (3) report file，參賽隊伍必須依照自己的設計內容，撰寫 report.000 檔，以方便主辦單位進行評分，report.000 的格式如下圖所示。(report 檔以後三碼序號表示版本，若繳交檔案更新版本，則新版的 report 檔的檔名為 report.001，依此類推)

表三、繳交檔案

<b>RTL category</b>		
<i>Design Stage</i>	<i>File</i>	<i>Description</i>
N/A	N/A	Design Report Form
RTL Simulation	*.v or *.vhd	Verilog (or VHDL) synthesizable RTL code
<b>Gate-Level category</b>		
<i>Design Stage</i>	<i>File</i>	<i>Description</i>
Pre-layout	*_syn.v	Verilog gate-level netlist
Gate-level Simulation	*_syn.sdf	Pre-layout gate-level sdf

### report 檔

FTP account:	E22xxx, FTP 帳號
Level:	A/B/C/D 設計完成等級
cycle :	600000, 三組 testbernc 中，模擬 cycle 最長的 cycle 數
Synthesis area :	10000，合成 report 的 cell area
--- RTL category---	
HDL simulator :	ncverilog/vcs, 使用之 HDL 模擬器名稱
RTL filename :	JAM.v, RTL 檔案名稱以及使用到的子模組檔案...
--- Pre-layout gate-level ---	
gate_level filename:	JAM_syn.v, gate-level 檔案名稱
gate-level sdf filename:	JAM_syn.sdf, sdf 檔案名稱

## 附錄 D、檔案壓縮整理步驟

當所有的文件準備齊全如表三所列，均需要提交至 TSRI。請按照以下的步驟指令，提交相關設計檔案，將所有檔案複製至同一個資料夾下壓縮，步驟如下：

1. 建立一個 result\_xxx 資料夾。其中“xxx”表示繳交版本。例如“000”表示為第一次上傳；“001”表示為第二度上傳；002 表示為第三度上傳，以此類推...。  
    > **mkdir result\_000**
2. 將附錄 C 要求的檔案複製到 result\_xxx 這個目錄。例如：  
    > **cp JAM.v result\_000**  
    > **cp JAM\_syn.v result\_000**  
    > **cp report.000 result\_000**  
    .....
3. 執行 tar 指令將 result\_xxx 資料夾包裝起來，tar 的指令範例如下：  
    > **tar cvf result\_000.tar result\_000**  
    執行完後應該會得到 result\_000.tar 的檔案
4. 使用 ftp 將 result\_xxx.tar 上傳至 TSRI 提供的 ftp server，評審將以最後上傳的設計檔編號進行評分作業。

上傳之 FTP 需切換為二進制模式(binary mode)，且傳輸埠均設為 21(port:21)。

ftp 的帳號和密碼在賽前已用 email 寄給各參賽者。若有任何問題，請聯絡 TSRI

FTP site1 (新竹半導體中心)：iccfpt.tsri.org.tw (140.126.24.18)

FTP site2 (南區半導體中心)：iccfpt2.tsri.org.tw(140.110.117.9)

EDA Cloud內請見開啟terminal時訊息

5. 若需要繳交更新版本，請重覆以上步驟，並記得修改 tar 檔的版本編號，因為您無法修改或刪除或覆蓋之前上傳的資料