IC Contest 2022

Table of Contents

Page

[Question: 1](#_Toc130160903)

[My Answer: 1](#_Toc130160904)

[重點 1](#_Toc130160905)

[整體概念 2](#_Toc130160906)

[字典排序法 3](#_Toc130160907)

[八個週期比較器 4](#_Toc130160908)

[Data Flow Graph 5](#_Toc130160909)

[RTL sim Result: 6](#_Toc130160910)

[Design Compiler Area 8](#_Toc130160911)

[Gate-Level Sim Results 8](#_Toc130160912)

# Question:

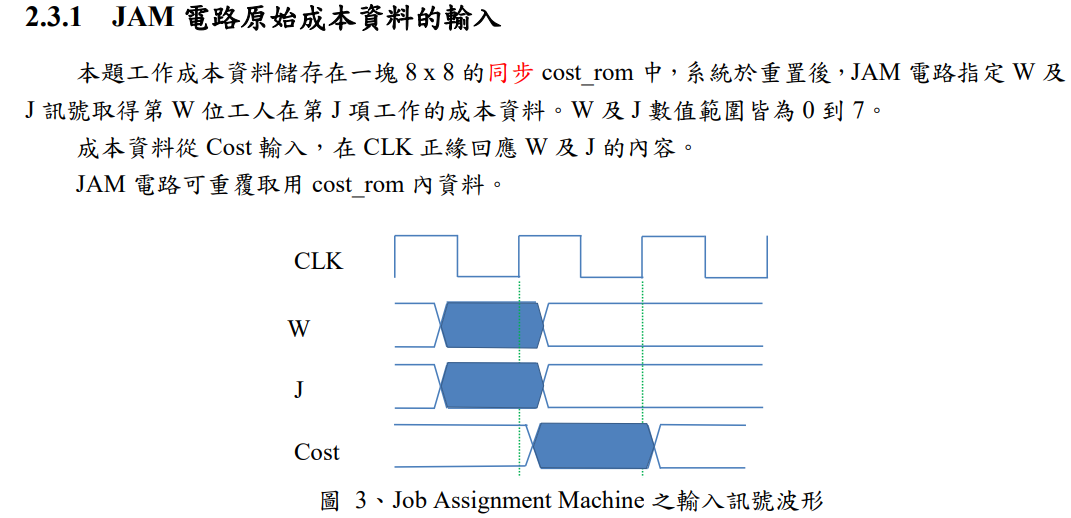
Detail of the Question from the PDF file.

# My Answer:

## 重點

題目要求列出所有可能情況，並找出最低成本以及最低成本的組合數量。本參考解答是利用題目上的字典排序法當作演算法作答。

首先在解題前要先了解題目意思，題目有提到:



也就是說，每一個週期只能拿到一筆資料，一次只能傳出一筆由J,W組合而成的數字，再收到一筆cost的資料。也就是說，要拿完八筆資料之後再比較資料後再判斷是否為最小成本組合。所以，每一次做上面的排序法，加上之後判斷最大最小等等運算，要在八個周期之內完成。

## 整體概念

整體概念是要在八個周期內每一周期輸出一個W,J並且累計八個周期內的Cost與上一周期的累計值比大小，小的留下來，計數器設為一，如果累計值一樣，計數器加一。

全部的組合情況是8!，在最後排序到[7,6,5,4,3,2,1,0]時，下一周期要拉高Valid並輸出最小成本和最低成本組合，每一個組合有八個周期可以運算，所以在RST後最快可以完成的時間是

最快能夠在322560周期之後找到最小成本以及最低成本組合。

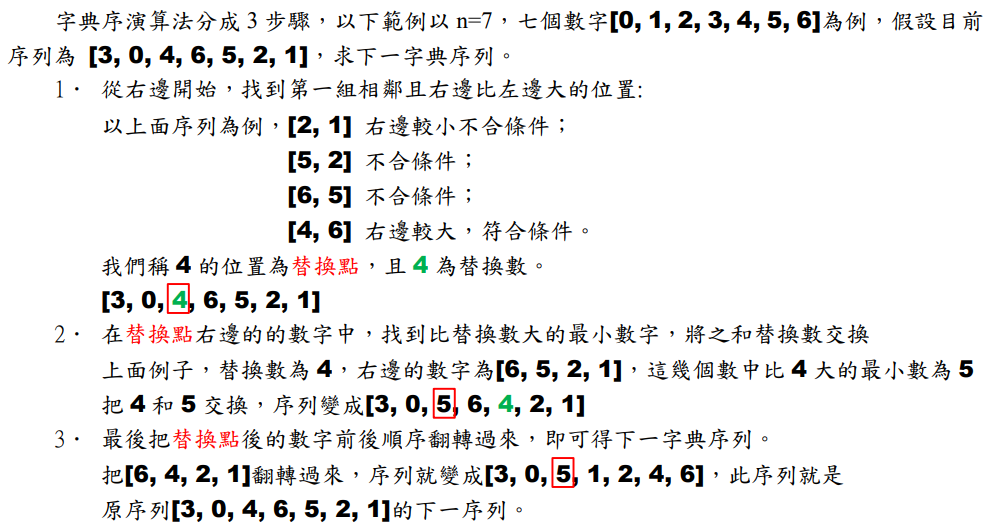
所以整個設計分為兩個區塊:

1. 設計一個電路要能夠輸出每一種序列可能，輸出最後一個序列為[7,6,5,4,3,2,1,0]時輸出一個last訊號，表示這是最後訊號。
2. 設計一個電路能夠累計八個周期內的cost，並與上一個累計值比大小。

了解設計目標後，再來看如何實現細部的演算法。

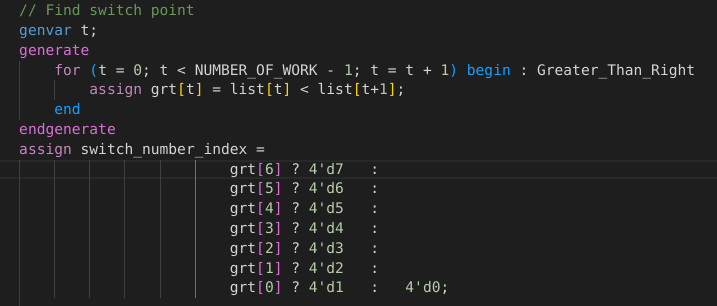
## 字典排序法

首先，字典排序法:



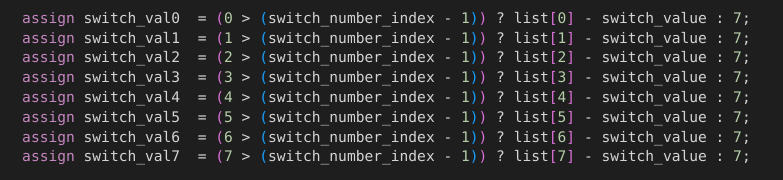
字典排序法一開始的組合是[0,1,2,3,4,5,6,7]，最後一組的組合是[7,6,5,4,3,2,1,0]。

第一步驟:由右邊開始，找到序列中第一組相鄰且右邊較大的位置。用七個比較器去比八個數字大小，Com[0,1], Com[1,2]…Com[6,7]，題目要求由右邊開始找比左邊大的數值，所以Com[6,7]這一組會最優先當作替換點的比較器數值。之後再依序排下去，當發現Com中沒有一組為1時，表示序列右邊數字都比左邊數字大，就代表所有情況都排序過了。不過我是直接判斷輸出端的序列是否等於最後一組。



第二步驟:找到替換點右邊數字中，比替換點大的最小數字，兩數字交換位置。這裡有兩個點要注意

1. 要是替換點右邊的數字，又同時是右邊數字中最小的數字。再比較小時，要找替換點右邊的數字，所以不是右邊的值，就把它設為最大，等等在比小時，它就不會被選到。
2. 再來是找比替換點大的最小數字，比替換點大很簡單，就直接比，但比的方式是用減法，把在右邊的值通通減去替換點的值。這樣一來比替換點的值大的數字會變成正的很小，但比替換點的值小的數字會因為overflow變成正的很大。



用比小的方式找到替換點右邊數字中，比替換點大的最小數字之後就把兩數交換位置。

第三步驟:把替換點後的數字順序倒過來。

設計上面的電路時，可以分很多cycle慢慢比較大小，不用同時在同一cycle做完，做計算時要注意資料之間的傳遞順序以及時序。

## 八個週期比較器

題目要求輸出最小成本組合數(MatchCount)，以及最小成本(MinCost)，所以需要找到最小成本的比較電路。

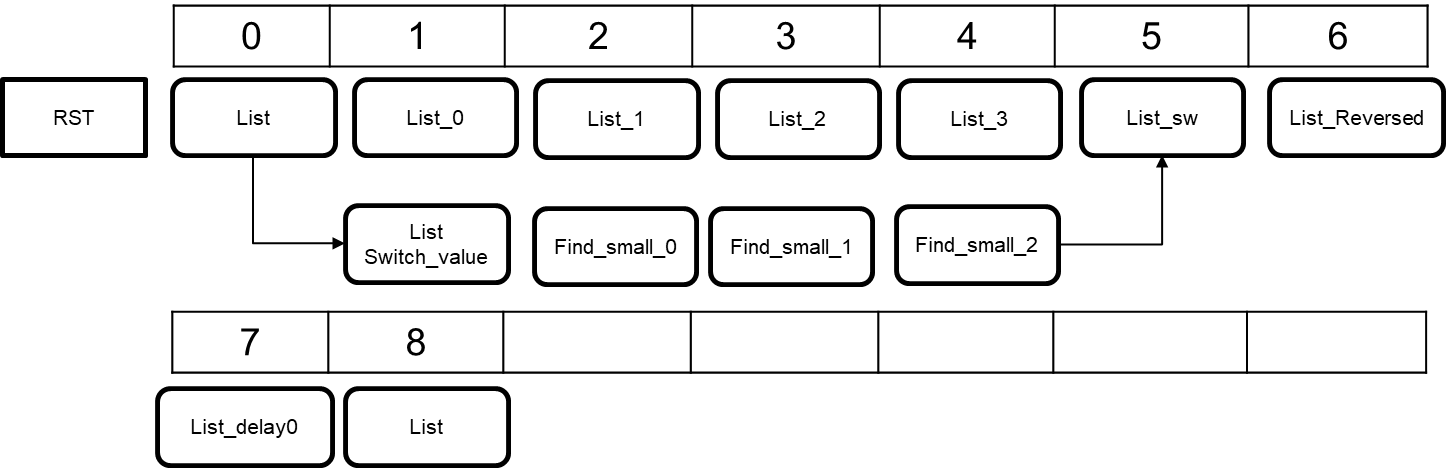
1. 設計一個累計八周期的累計器(Accumulator)當作目前八周期內的成本
2. 找目前為止的MinCost，所以每一次Accumulator累計完就跟目前為止的MinCost比小:

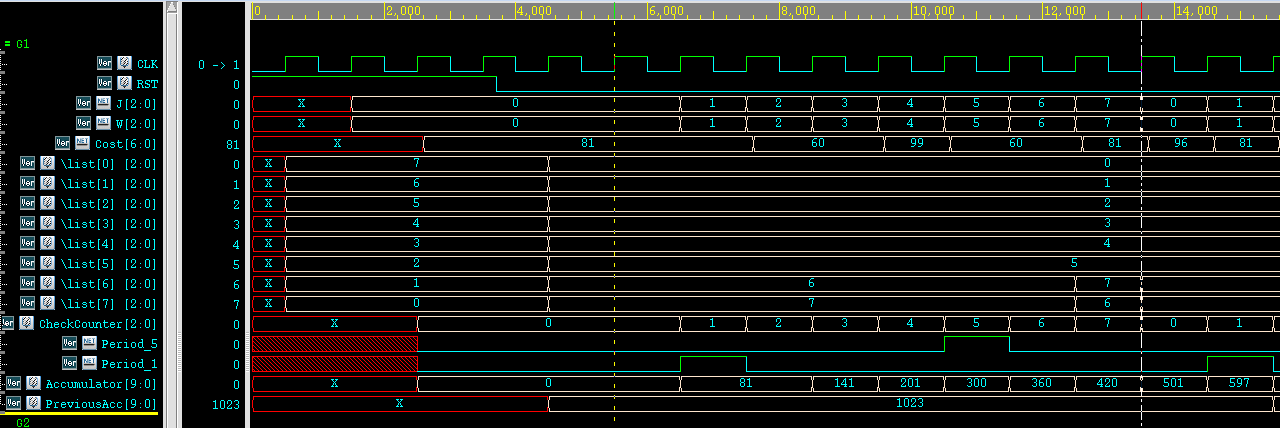
如果Accumulator比較小就更新MinCost，並將MatchCount設為1

如果Accumulator一樣，MatchCount+1

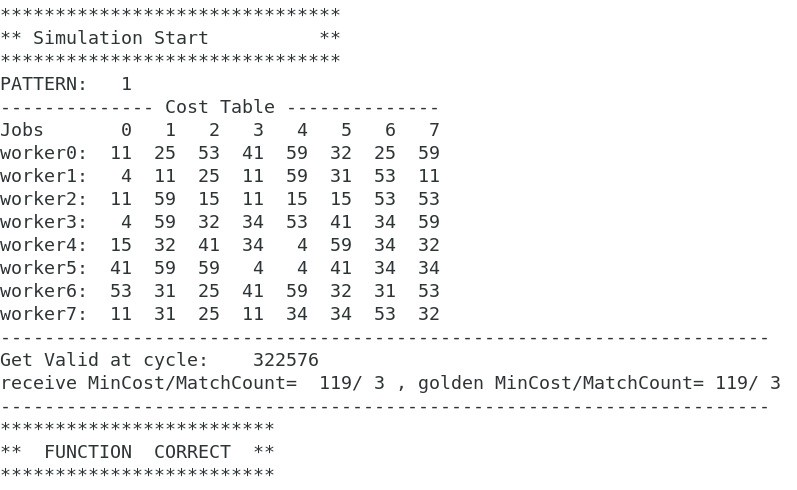
設計時注意，累計八個週期的Accumulator，再要被更新值之前，要將Accumulator存到另一個暫存器PreviousAcc，因為如果在拿到八個周期內的累計值時比大小，會因為下一筆cost又進來，所以Accumulator被更新為新一筆Cost而非上八周期內的累計值。

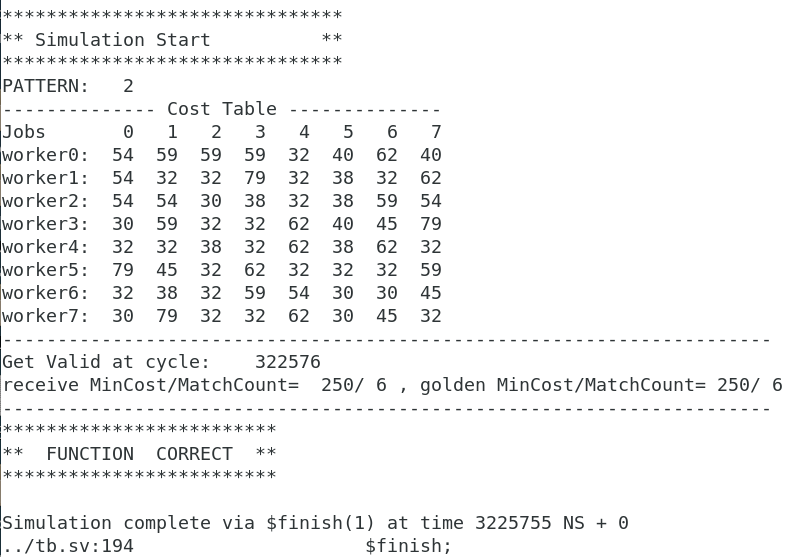
## Data Flow Graph

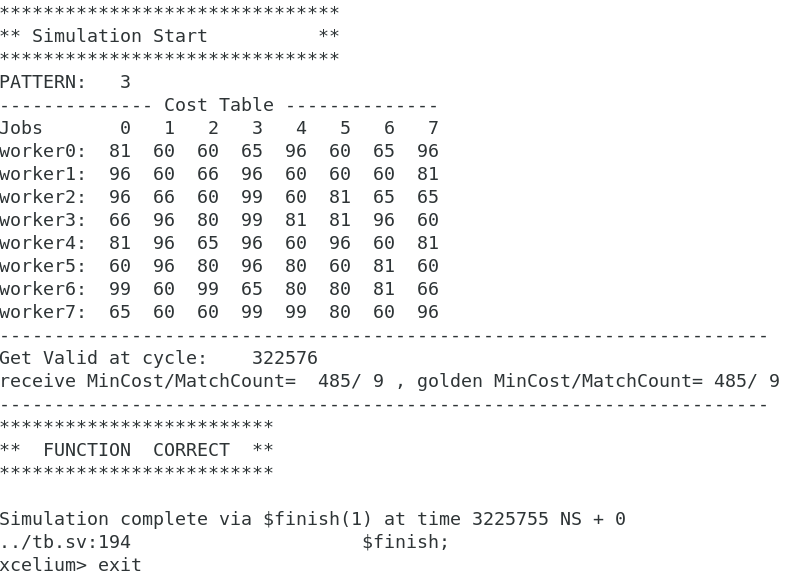


1. List為input也是output，在傳輸到第八cycle時要回傳數值，當作下一次開始的序列。
2. 我是分成三個cycle來找最小值，反正cycle數很多，慢慢比。
3. 由於分三個cycle在比大小，原本的數值也要儲存，這樣在字典排序法第二步驟，要交換數字時才能知道要跟誰換。
4. 換完下一cycle把替換點後的數字反過來。就是下一序列的開始。
5. 由於要八個cycle當作一周期，算完沒事還要延遲一周期捕到八周期。

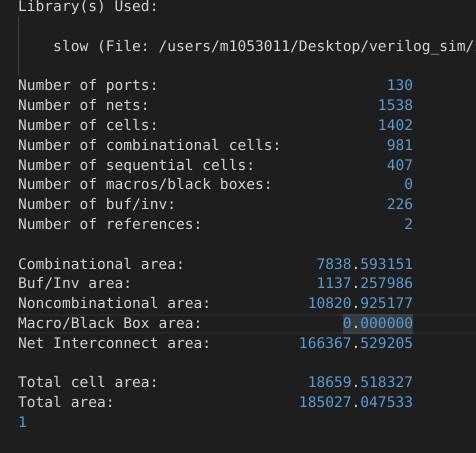
## RTL sim Result:







## Design Compiler Area



## Gate-Level Sim Results

