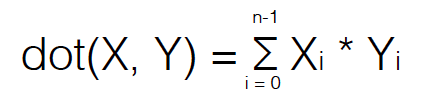
DLD – lab03 – Vector Inner Product

104020002 理雙19 周明曄

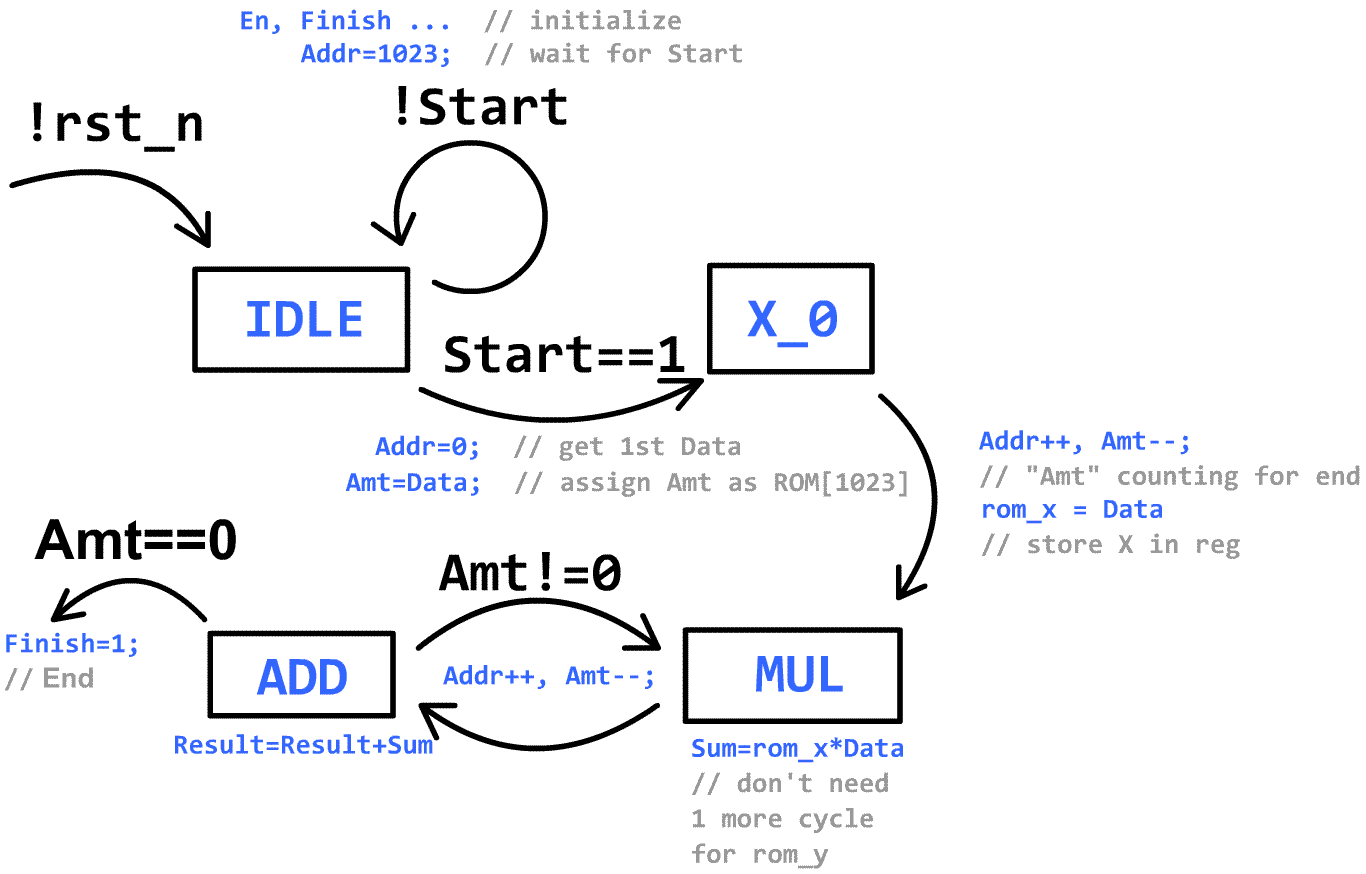
一、目的以及流程圖

※計算內積及其加總

1. 學會使用FSM來操作流程

2. 學會使用時脈clk及d\_ff操作sequential電路

二、Finite State Machine



IDLE:

1. 所有變數的初始，並等待Start拉起，即開始計算。

2. 等待Start拉起的這幾個cycle裡，會向ROM要取存放在[1023]的資料數，以方便得知計算是否結束。(存放在Amt中，做為終止條件)

3. Start拉起之後，會在此cycle中要取第一個X參數，並在下一個cycle得到。(在Addr的flip-flop中存放index,[0])

X\_0:

1. 得到X參數，存放在rom\_X中，並要取第一個Y參數。(Addr++, Amt--)

2. 無條件進入MUL

MUL:

1. 得到Y參數，並要取第一個X參數。(Addr++, Amt--)

2. 將rom\_X與要取到的Y用ALU乘起來存在Sum裡。

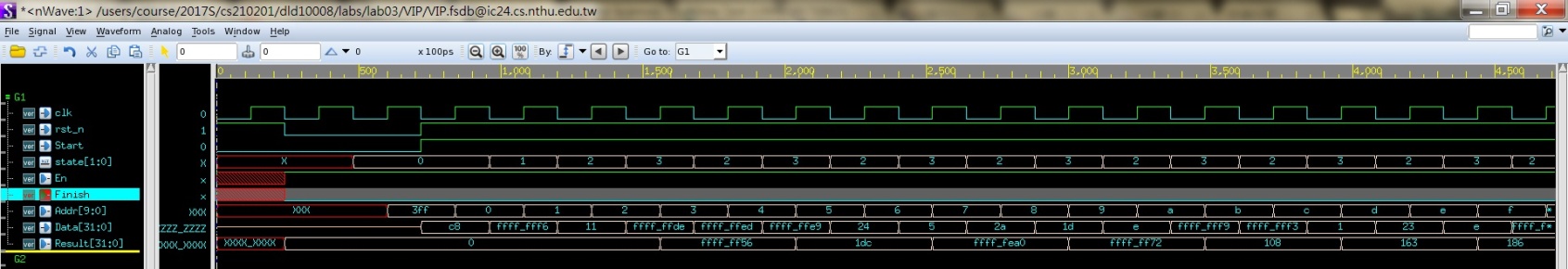
ADD:

1. 將目前的Sum跟上個cycle的Result加在當前cycle的Result。

2. 檢查Amt是否等於0。

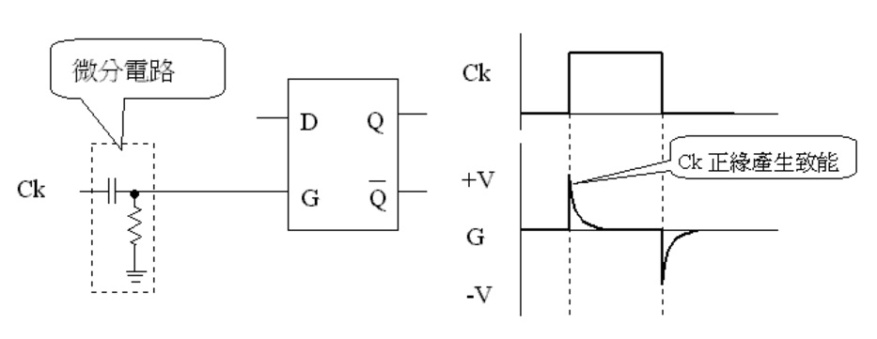
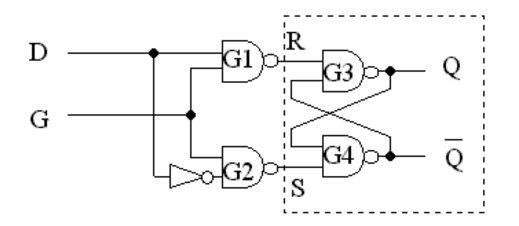
若是，則結束計算，使Finish為1，在下個cycle檢查答案正確與否。

若否，則繼續要下一個X參數，並重回MUL state。



三、Synthesis

D flip-flop / D Latch :



※左為flip-flop，右為Latch。

※圖片來源: http://www.gauss.com.tw/logic/ch9/index9.htm

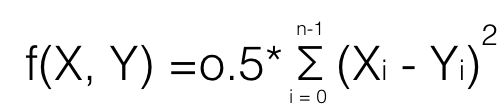
兩者差別在於，觸發的訊號，只要觸發存放資料的訊號為任意訊號的都可以被稱做Latch，而flip-flop的觸發訊號，一定要是時脈訊號。

我們合成reg得到Latch的時候，是因為我們在此cycle並沒有對此參數做任何指定或描述，進而讓合成軟體認為我們在未來將會用到此參數。但卻沒有指定，因此用非時脈觸發的型式來存儲，也就是Latch。

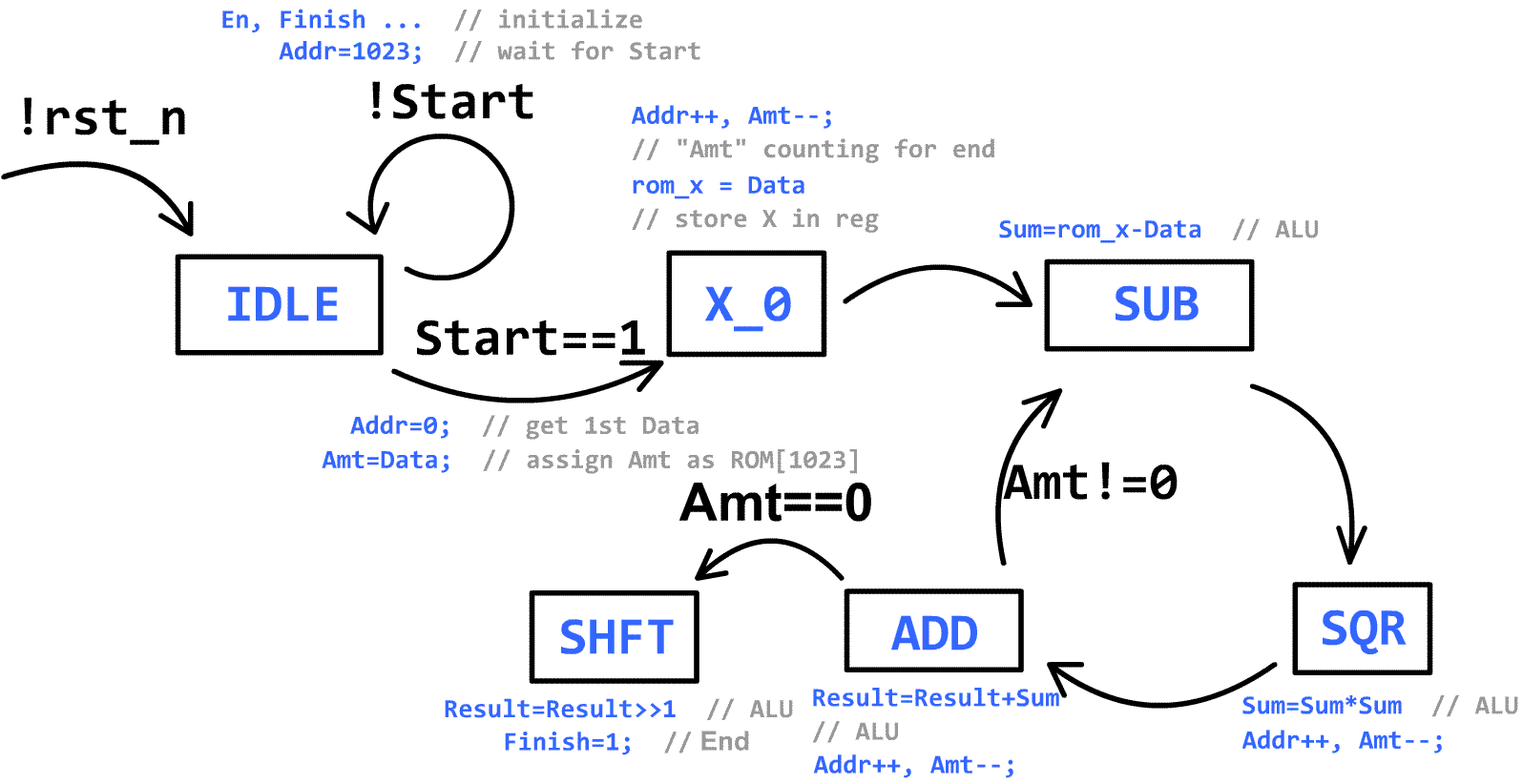
這種情況通常是發生在使用if卻沒有給定else，case並沒有討論完全，或是遺漏了某些參數的狀況。因此在此次lab中，只要先定義好d flip-flop的元件，包括rst情況，以及一個cycle中的清況處理(有指定新值及指定新值，沒指定則指定原值)，再把在FSM會用到的所有參數，全都丟到d flip-flop裡，就不會合出latch了。

※一開始沒有做好細節，Latch使我非常頭痛，不過後來我好像也沒有寫好沒指定新值則給定原值的部分，卻沒有合出Latch，覺得問號???

四、Bonus



FSM:

IDLE:

同VIP。

X\_0:

同VIP，進入SUB state。

SUB:

使用ALU將rom\_X與要到的Data(Y)相減，無條件進入SQR。

SQR:

將當前sum與自己相乘，存於下一個cycle會得到的sum裡，並要下個X。

ADD:

將當前sum與Result相加後存於下一個cycyle的Result。

檢查Amt是否為0，若是，進入SUB繼續循環；反之則結束，進入SHFT。

SHFT:

使用ALU右移一個bit，相當於乘0.5、除以二。

並且將Finish拉起，En放下。