

# HW4 Report

電機四 孟妍 b06901066

## (a) Simulation

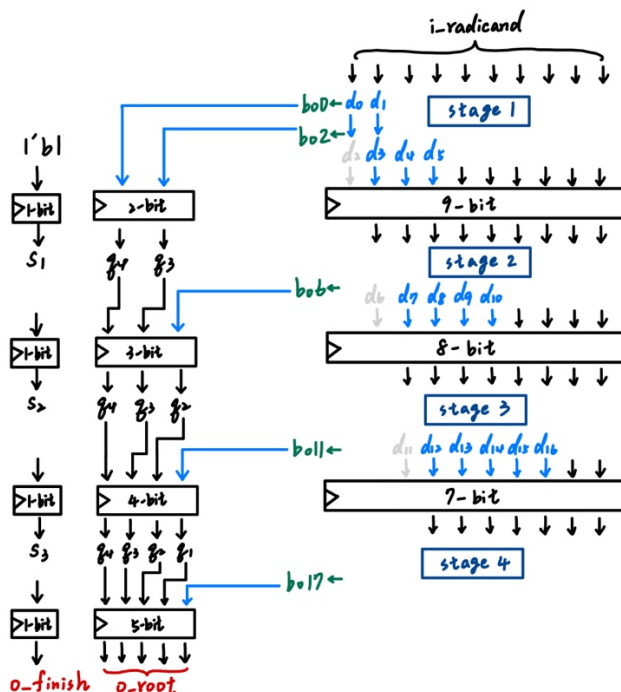
- minimum clock cycle: 3.2 ns / transistors: 1826
- strategy: long division + pipeline
- simulation result:

Simulation passed	
Summary	
Clock cycle:	3.2 ns
Number of transistors:	1826
Total excution cycle:	1003
Correctness Score:	40.0
Performance Score:	5860729.6

## (b) Circuit diagram

strategy: long division + 4 stage pipeline

part1. pipeline structure



stage1 會得到 o\_root 的最左邊兩個 bit, 其他三個 stage 就是每個 stage 會新算出一個 o\_root 的 bit。

上圖左到右分為三個部分：flag, quotient, input

- (1) flag 的部分需要 4 個 1 bit d flip-flop, 從 reset 結束後開始 propagate 1'b1 在 4 個 clock(4 stage)之後把 o\_finish 拉起。

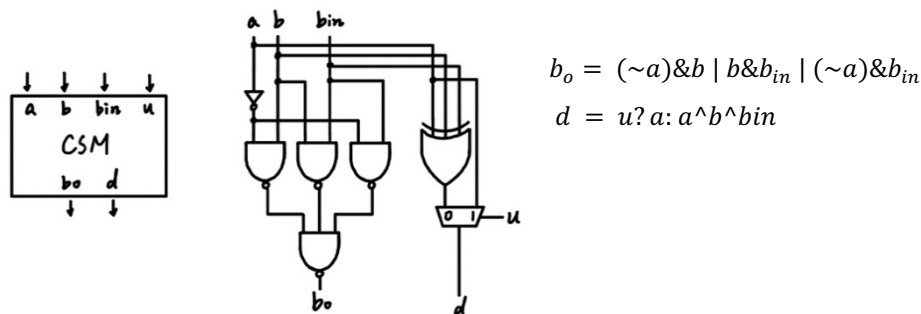
(2)quotient 的部分用了 4 個 flip-flop，每到下個 stage 就多一個 bit，因為每個 stage 會多新增一個 bit。

(3)input 的部分就是紀錄 i\_radicand 的變化，但因為我們最後不需要 remainder，所以這邊只需要三個 d flip-flop 就可以了，最後一個 stage 的結果不需要。這裡的三個 d flip-flop 則是每到下個 stage 少一個 bit，因為到下個 stage 之後前一個 stage 最左邊的 difference 不需要用到。

## part2. basic components

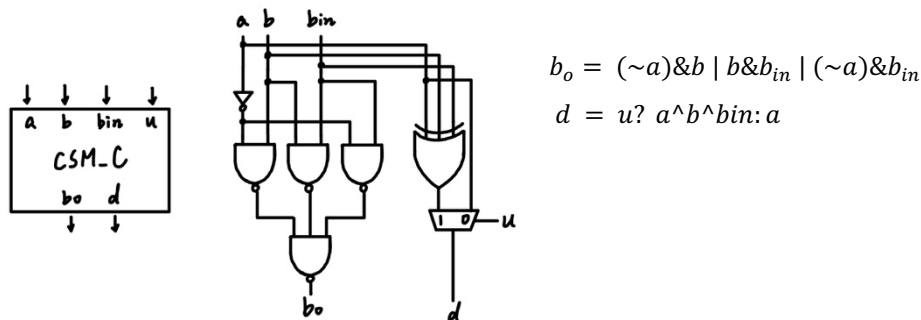
整個 pipeline 大致就是用以下的 components 和 flip-flop 完成

一個 CSM block 就是一個 controlled subtractor + multiplexer，架構如下：

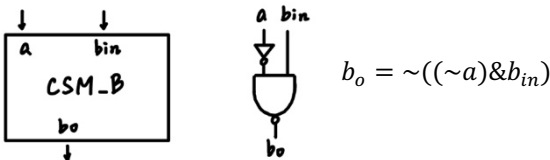


因為後來有做一些優化，所以整個 circuit 用了以下幾種 CSM block 的變形：

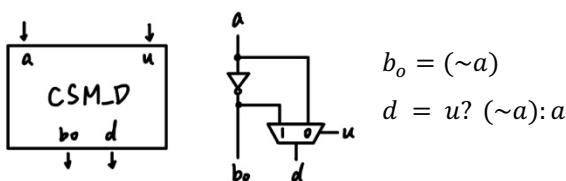
(1) CSM\_C: same as the origin CSM block only with u inverted



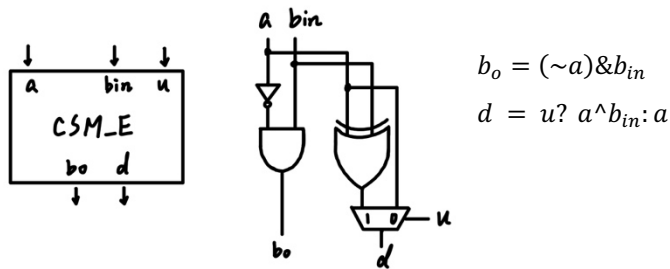
(2) CSM\_B: CSM block with  $b = 1'b0$  and  $u = 1'b0$ , with  $b_o$  inverted and without  $d$



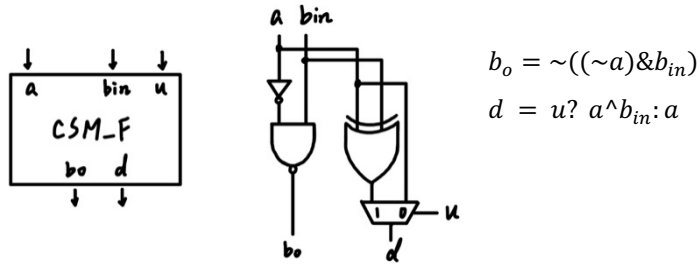
(3) CSM\_D: CSM block with  $b = 1'b1$  and  $bin = 1'b0$



(4) CSM\_E: CSM block with  $b = 1'b0$

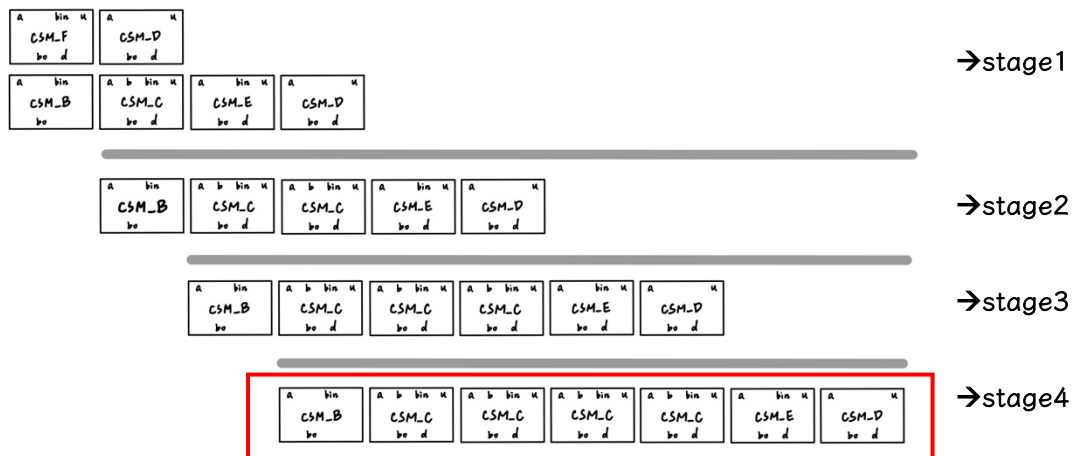


(5) CSM\_F: similar as CSM\_E only with  $b_o$  inverted



part3. circuit of each stage

overall structure

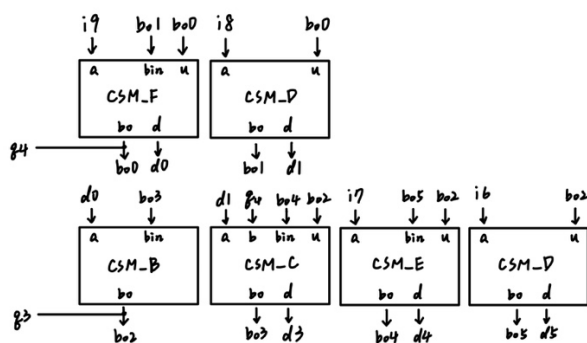


critical path 為 stage4 的 7 個 block 串接

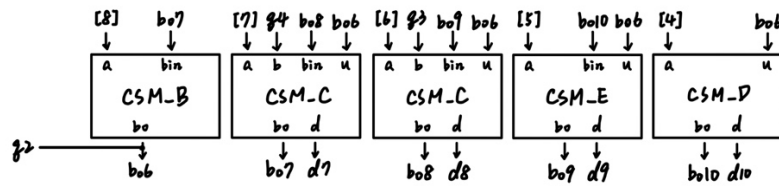
notations:  $iX = i\_radicand[X]$   $boX = bo$  of blockX  $dX = d$  of blockX

$[X] =$  input index of current stage

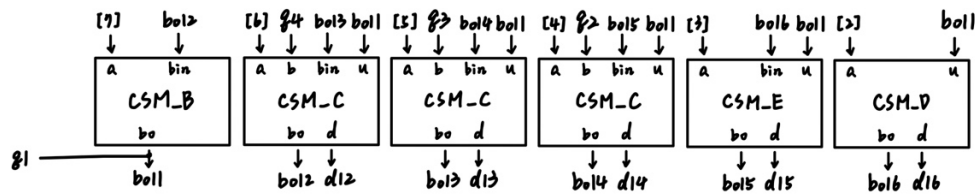
(1) stage1



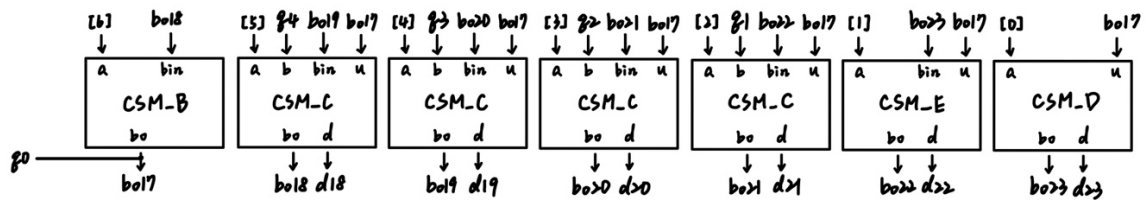
(2) stage2



(3) stage3



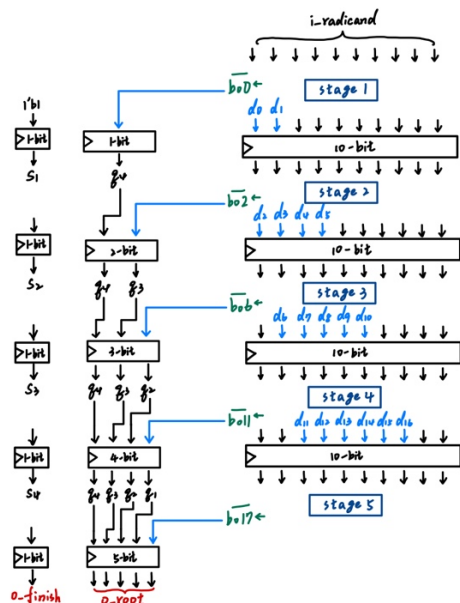
(4) stage4



(c) Discussion

[part1. pipeline](#)

因為助教說了長除法+pipeline，所以我就朝著這個方向做。一開始先做了最直觀 5 stage pipeline，一個 stage 就出一個 quotient bit，如下圖：



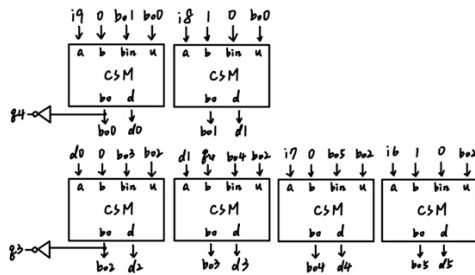
但這樣的問題就是前面的 stage 會跟後面 stage 的 latency 會差太多，多花一個 cycle 非常不優，因此就把 stage1 跟 stage2 merge，這樣 critical path 仍然是最後一個 stage，同時少一個 cycle 也少了很多組成 flip-flop 的 transistor。後來有再試了把 stage1-3 merge，看看 A\*T 的 tradeoff，因為少一個 stage 可以少蠻多 flip-flop 的 transistors，但結果是 4 stage A\*T 比較小，所以最後就還是用 4 stage 的架構。

## part2. optimization

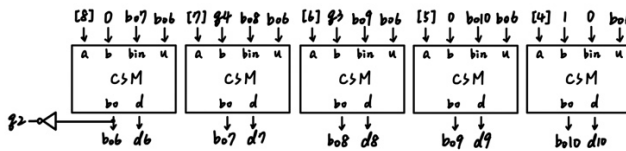
原本的 circuit 就都用同樣的 CSM block 做，但連接了每個 stage 完整的 input/output 後發現有一些 block 的某些 input 會是常數，或是其中的一個 output 不會被用到，所以就細分成了幾種 block，可以大幅縮短 critical path

origin:

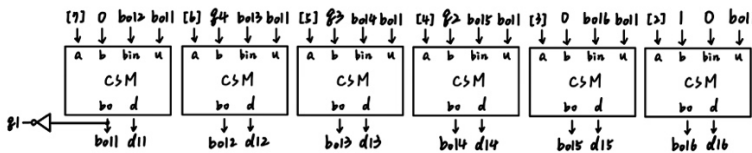
(1) stage1



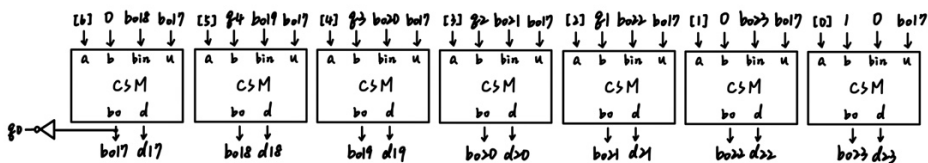
(2) stage2

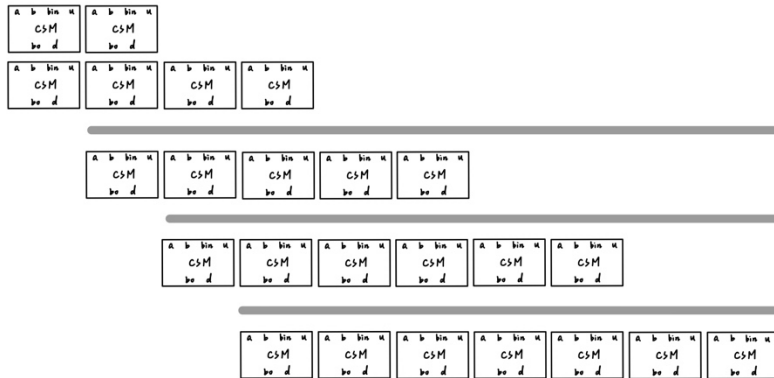


(3) stage3

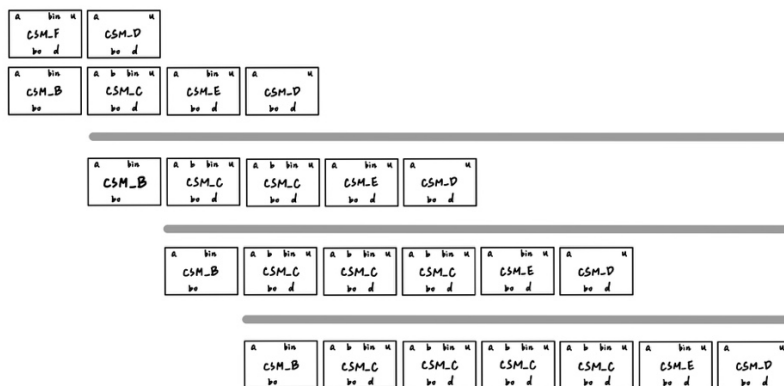


(4) stage4





modified:



後來有再做一些更細微的改變來降低  $A^*T$ ，像是原本每個 stage 最左邊的 block 的 borrow out(bo)需要再經過 inverter 才會是 quotient bit，所以後來就改成每個 stage 最左邊的 block 裡的 AND gate 改成 NAND gate，讓 bo 輸出直接是 quotient bit。

因為每個 stage 最左邊的 bo 會是同個 stage 所有 block 裡的 multiplexer select condition，若 quotient(u)為 0 選擇原本的 input 反之選擇相減的結果，因此改變最左邊的 block 的 bo 只需要讓 multiplexer select 的條件相反即可，不會增加 transistor 或拉長計算時間，如此 critical path 可以再降低一點以及再減少一些 transistor。

另外就是原本 pipeline 每個 stage 的 input 都會是 10 bit d flip-flop，但後來發現因為最後不用輸出 remainder 所以其實每到下一個 stage 就會有一些 bit 不會用到了，所以也減少每個 stage hold input 的 d flip-flop 的 bits。

最終完整的 circuit diagram 如(b)