

113 學年度

國立中山大學

課程名稱:硬體描述語言

**題目: Homework 1 Adder Designs Using Verilog  
Structural, Dataflow, and Behavioral Modeling**

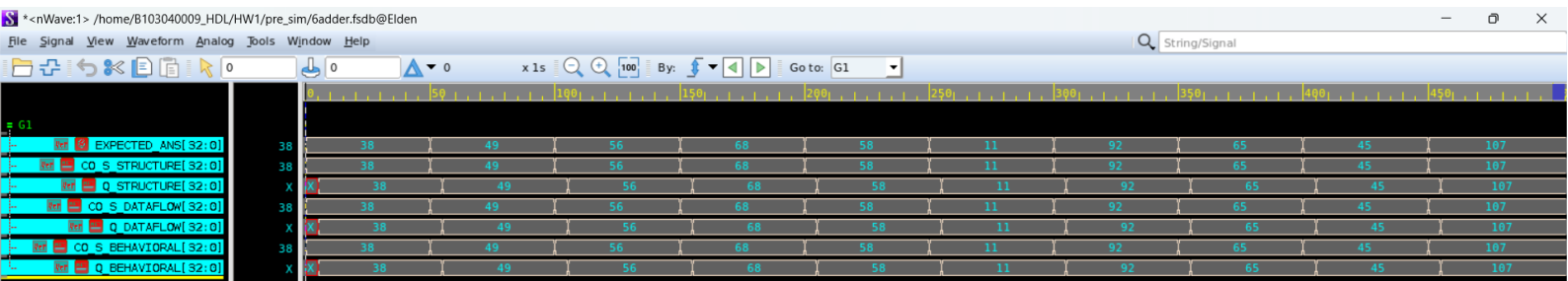
作業/成果報告/專題

授課教師: 蕭勝夫

學生學號/班級/姓名: B103040009/資工系/尹信淳

一、 模擬的波形圖

Pre-simulation:



### 三、 觀察(三種 modeling 之電路的數據/波型是否相同)

關於三種 modeling 的數據，不管有無 D flip flops，我發現 dataflow 與 behavior 的數據都相同，起初以為是做錯了，但後來上網查詢，發現像加法器這種簡單的設計是有可能合成出一模一樣的電路的。至於 structure(gate-level) modeling，我覺得因為是較明確地指定了電路設計後才丟進 design compiler，所以與較高層次的 dataflow, behavior modeling 經過 design compiler 合成出的電路在數據上有差異是合理的。

上述提到的這個數據差異，不管是專注在最小化 delay 時的 timing 或是最小化面積時的面積，用高層次的 modeling(dataflow or behavior)描述後，全權交給 design compiler 合成出的電路都好過自己寫 structure level modeling 不少。但是 power 方面卻是後者會有比較好的表現，我認為 design compiler 對於 power 方面是沒有在限制的，所以越是高層次的 modeling，design compiler 的發揮空間越大，犧牲的 power 數據就越多，造就越佳的 timing 或 area。

波型方面，有 D flip flops 的版本，因為在 CLK 信號正緣觸發時，新的數據才會顯現在 Q，所以在 A,B,C\_IN 改變後，要等到最近一個正緣觸發，正確的答案才會顯現在 Q。除此之外，六個波型都一樣，不管 pre-simulation 或 post-simulation 都是。

### 四、 心得

這次的作業中，首先我對於三種 modeling 以及 testbench(pre-sim&post-sim)有了實質的試作，可說是正式進入了硬體描述語言這門學問。此外，我也體驗了 simulator (synopsys vcs)，在這之前我都是在 Linux 上使用 Icarus Verilog 模擬，搭配 GTKWave 產出波型。vcs 在我看來是比較完善的模擬軟體，Icarus Verilog 感覺就不太可能做 post-simulation。

不僅如此，這次最難能可貴的經驗是使用了 synopsys design compiler，對 verilog code 進行 logical synthesis，首次拿到了所謂的 gate-level netlist。這讓我了解到 gate-level modeling 與 gate-level netlist 還是有差距的，我原先認為 gate-level modeling 跟 gate-level netlist 是一樣的東西，所以前者不需再做邏輯合成。但這次作業我觀察了 gate-level model 與其邏輯合成出的 netlist，是有差異的。前者會經過 technology mapping 產生後者，所以後者使用的是 cell library 中真正的硬體元件。此外，後者通常會是經過 delay-optimization, area-optimization 等等，得出的結果。簡潔地說，gate-level modeling 還是屬較高層次。