Power Optimization
(Clock Gating)

課程：算術處理器設計與實作、硬體描述語言

邱奕綸 chiu91523@gmail.com

大綱

OUTLINE

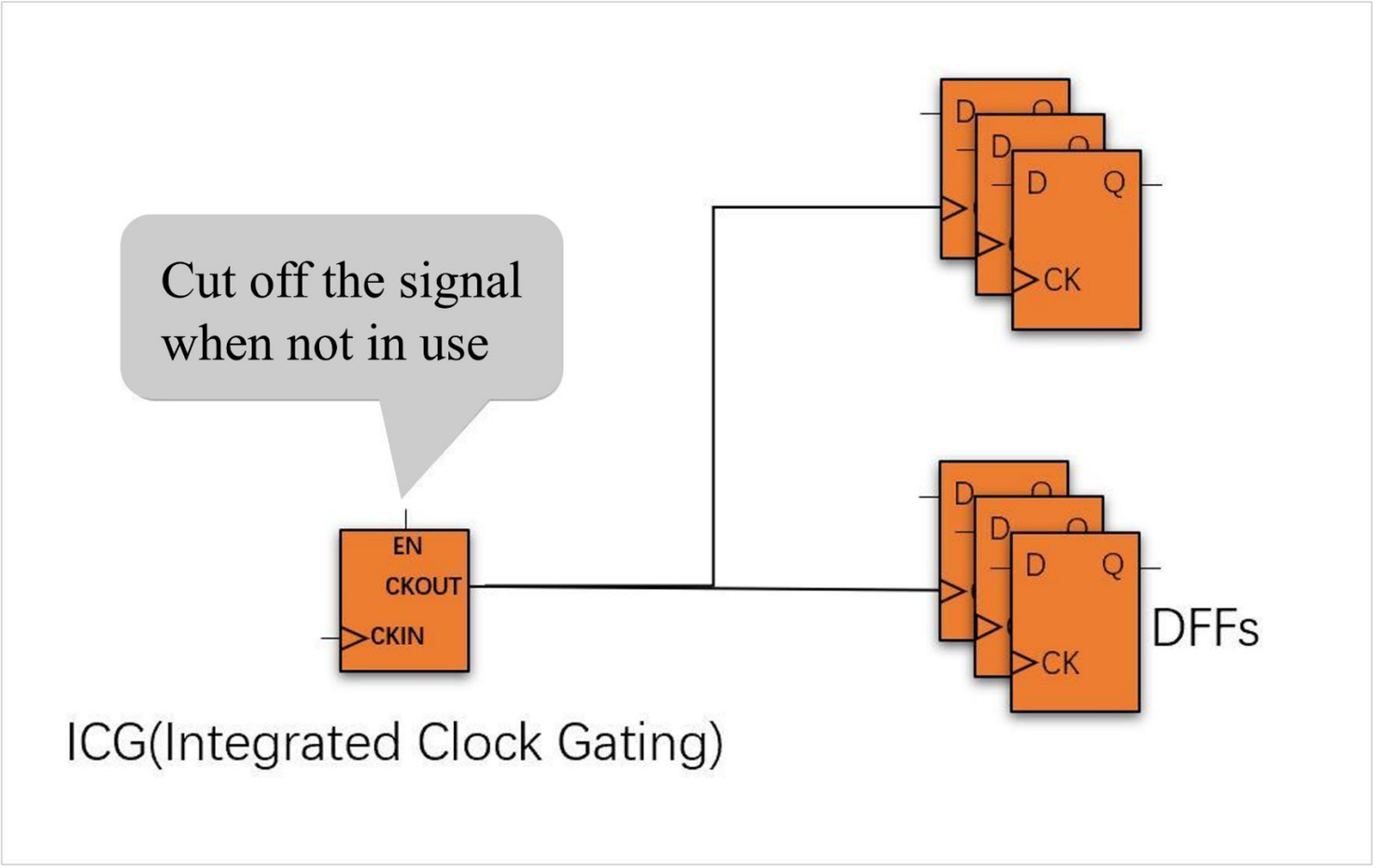
01 Introduction

02 Manual Methods

章節
Part 01 Introduction

Introduction

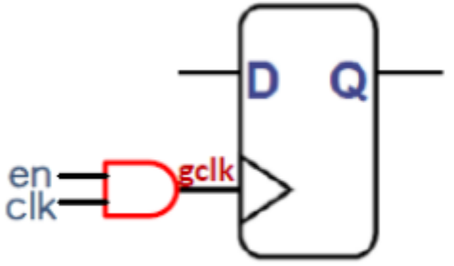
Clock Gating 介紹 (1/2)



透過Enable訊號來控制各DFF的Clock，不用時將Clock切斷(維持0)，從而減少翻轉訊號(Toggle Rate)，使動態功耗有所下降。

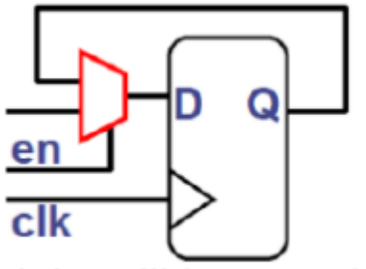
Clock Gating 介紹 (2/2)

```
// manual clock gating
module dff(Q, D, clk);
input D, clk;
output Q;
reg Q;
wire gclk, en;
// clock signal is from the output of AND
// glitch might cause extra clock edges
assign gclk = clk & en;
always @(posedge gclk)
   Q <= D;
endmodule
```



gclk might have glitches !!!
cause unexpected latching

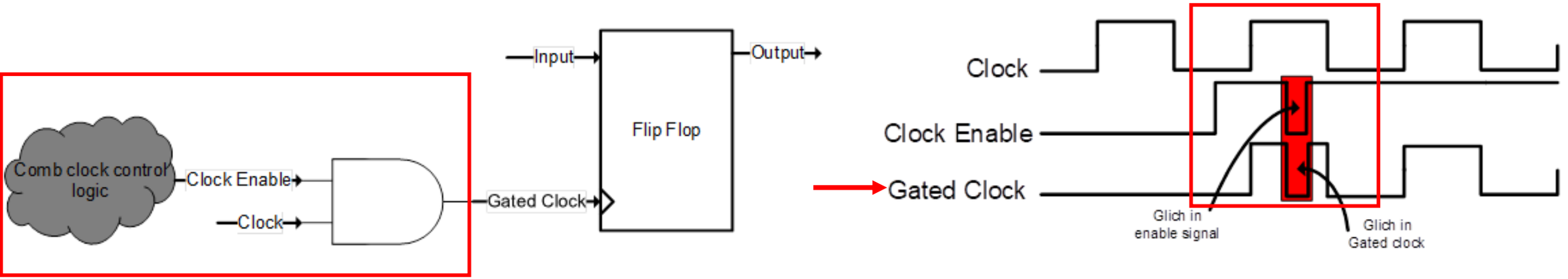
```
// automatic clock gating
module dff(Q, D, clk);
input D, clk;
output Q;
reg Q;
wire en;

// data input from MUX
always @(posedge clk)
if (en) begin
Q <= D;
end
endmodule
```



The clk port might still have switching power!
not efficiently reduce dynamic power

Introduction

Clock Gating (non-latch Version) (1/2)



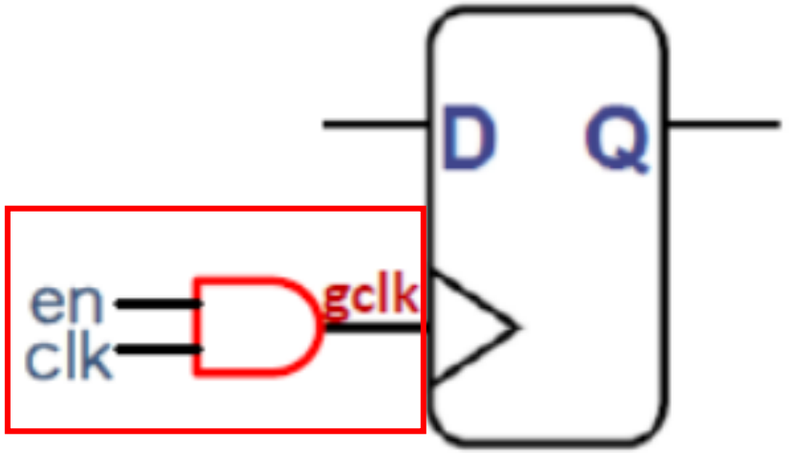
將Clock Enable(致能訊號)與Clock(時脈訊號)做AND並將訊號給予該DFF之Clock端即可達到Gating效果，如果使能訊號為0，Gated Clock將會維持0，該Gated DFF將不會有所動作，從而降低功耗。

如果只有AND去做Gating將會帶來一個問題，EN訊號通常為組合邏輯，因此當EN為1時Clock訊號才有用，如果EN在Clock為高準位時變為邏輯0將會有Glitch產生(如上右圖可以看到Gated Clock有窄波出現)，造成Data錯誤及功耗上升。

Clock Gating (non-latch Version) (2/2)

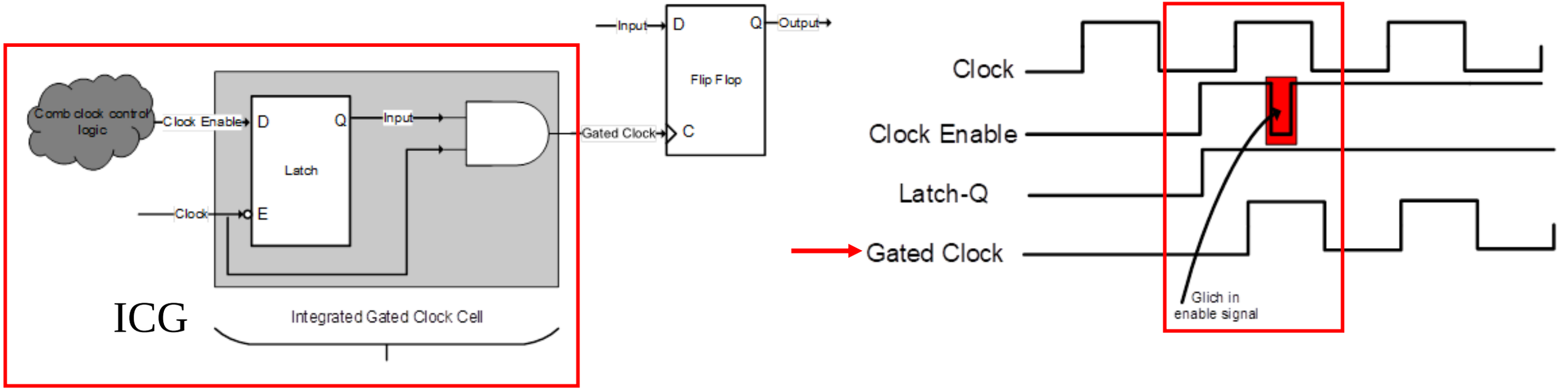
Clock Gating (non-latch Version) Verilog Code Example

```
module clock_gated_dff1 (Q, D, clk);
input D, clk;  output Q;
reg Q;
wire gclk, en;

assign gclk = clk & en;
always @(posedge gclk)
   Q <= D;
endmodule
```



non-latch Version代表只有AND Gate而已。

Clock Gating (Latch-Based Version) (1/2)

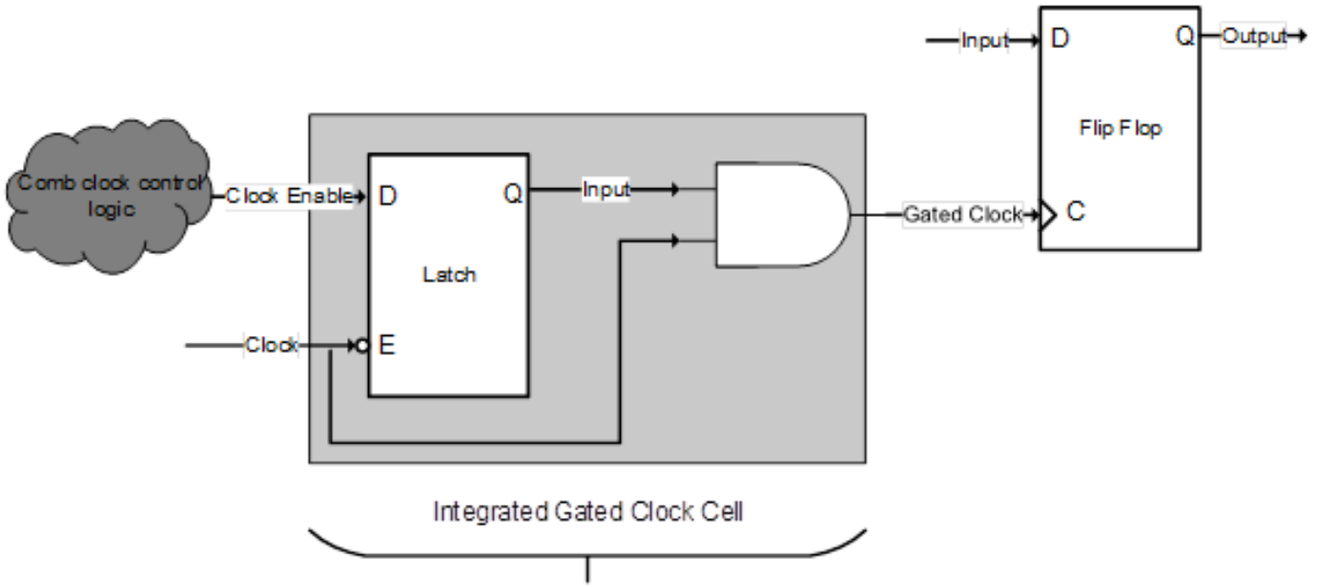


為了解決Glitch產生的問題，我們可以在AND Gate前加入Latch (Low-level sensitve)即可解決Glitch問題，使Gated Clock的波型為正確(無窄波問題產生)。

Clock Gating (Latch-Based Version) (2/2)

Clock Gating (Latch-Based Version) Verilog Code Example

```
module safe_clock_gated_dff (Q, D, clk);
input D, clk;  output Q;
reg tmp, Q;

always @ (CLK or EN) // latch
  if (!CLK)  tmp = EN;
assign gclk = tmp1 & CLK;  // AND gate
always @(posedge gclk)  // CG FF
  Q <= D;
endmodule
```



Latch-Based Version代表有Latch及AND Gate形成之ICG Cell(如上圖)，RTL Code較為冗長，因此後面會教如何透過Tool(Design Compiler)將non-latch Clock Gating自動轉為Latch-Based Clock Gating。

章節
Part 02

Manual Methods

Manual Clock Gating (1/4)

Method 1：Auto Clock Gating (not recommend)

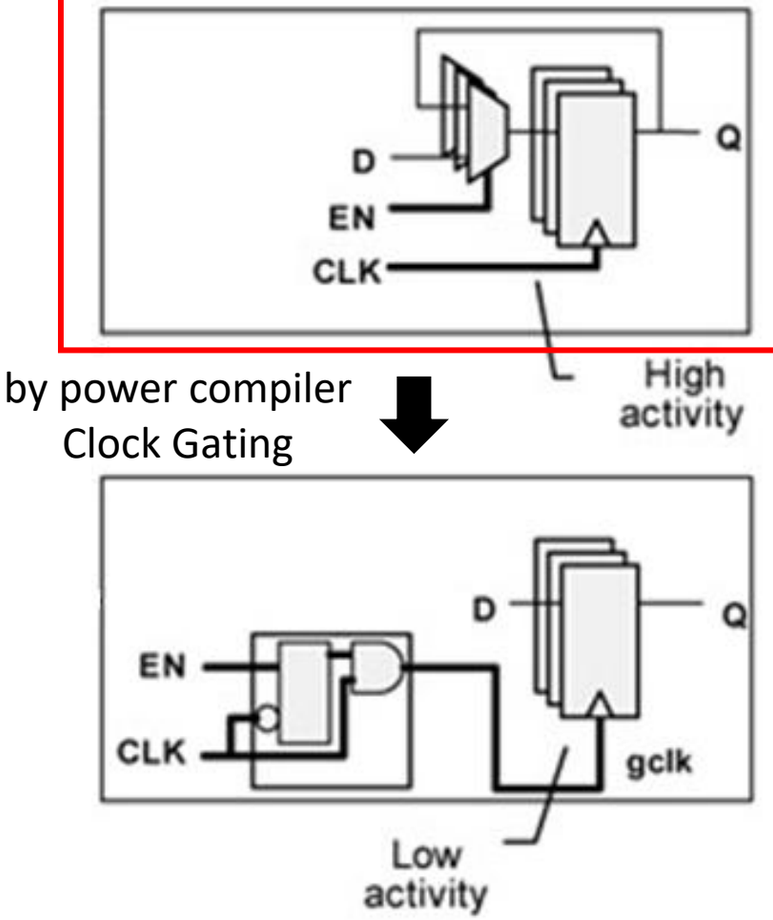
盡量不要用這種寫法，因為會產生如右圖上方電路。

RTL (Verilog Code):

```
always@(posedge clk or negedge rst_n) begin
    if(!rst_n)  Q <= 0;
    else begin
        if(en)  Q <= D;
    end
end
```

Design Compiler (TCL):

```
compile –gate_clock
```



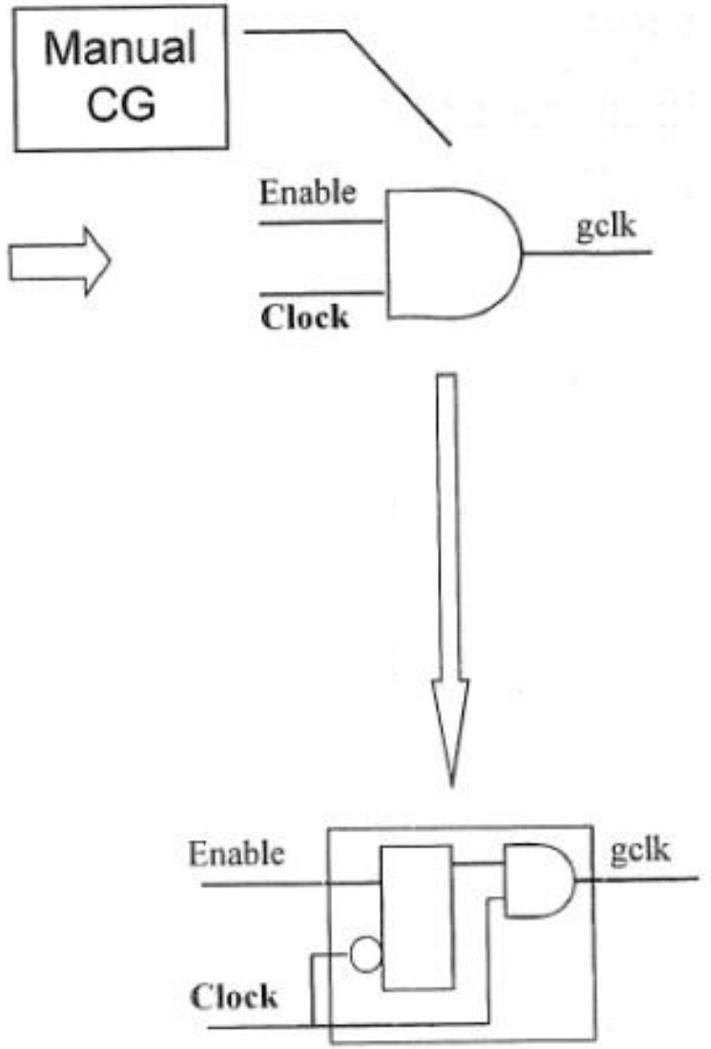
如果用以上寫法還是可以透過DC上方指令自動轉換成正確之Clock Gating電路。

Manual Clock Gating (2/4)

Method 2：Clock Gating (**recommend**)

RTL (Verilog Code):

```
assign gclk = clk & en;

always@(posedge gclk or negedge rst_n) begin
    if(!rst_n)  Q <= 0;
    else        Q <= D;
end
```



Design Compiler (TCL):

```
replace_clock_gates
compile
```

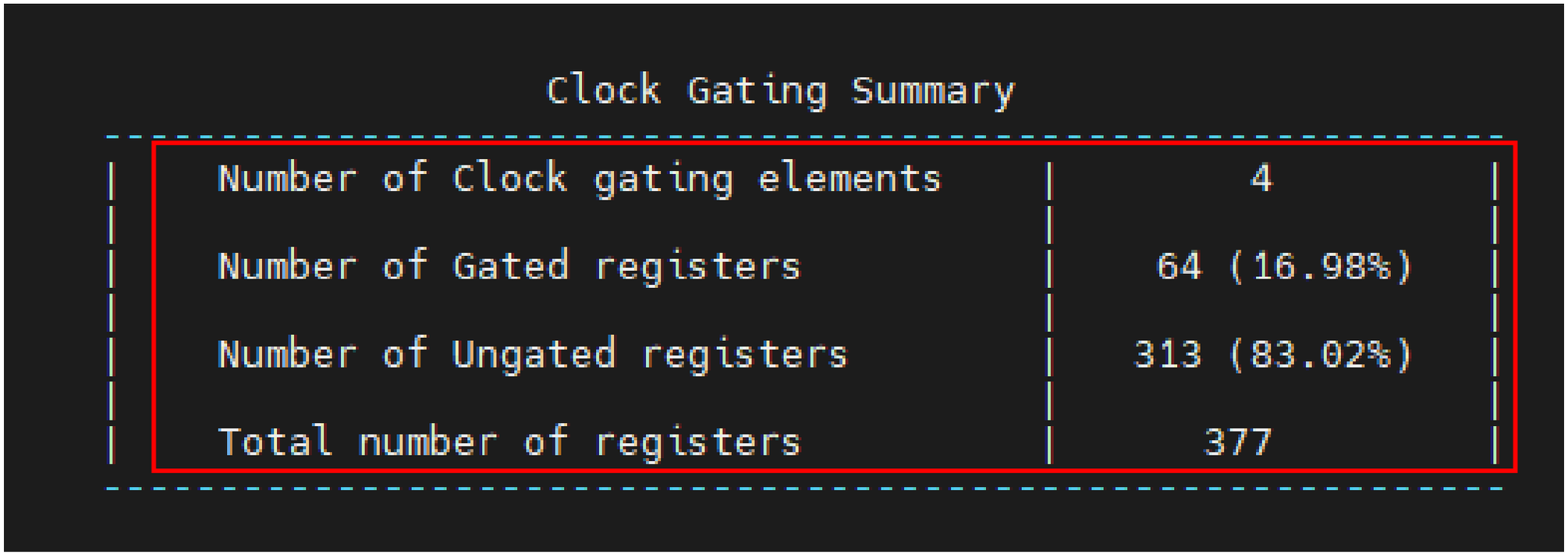
RTL使用(non-latch Version)寫法即可透過Design Compiler轉成Latch-based Clock Gating。

Manual Clock Gating (3/4)

Design Compiler (TCL):

```
report_clock_gating –gating_elements
```

可透過以上指令回報Gating DFF之數量



Clock Gating Summary

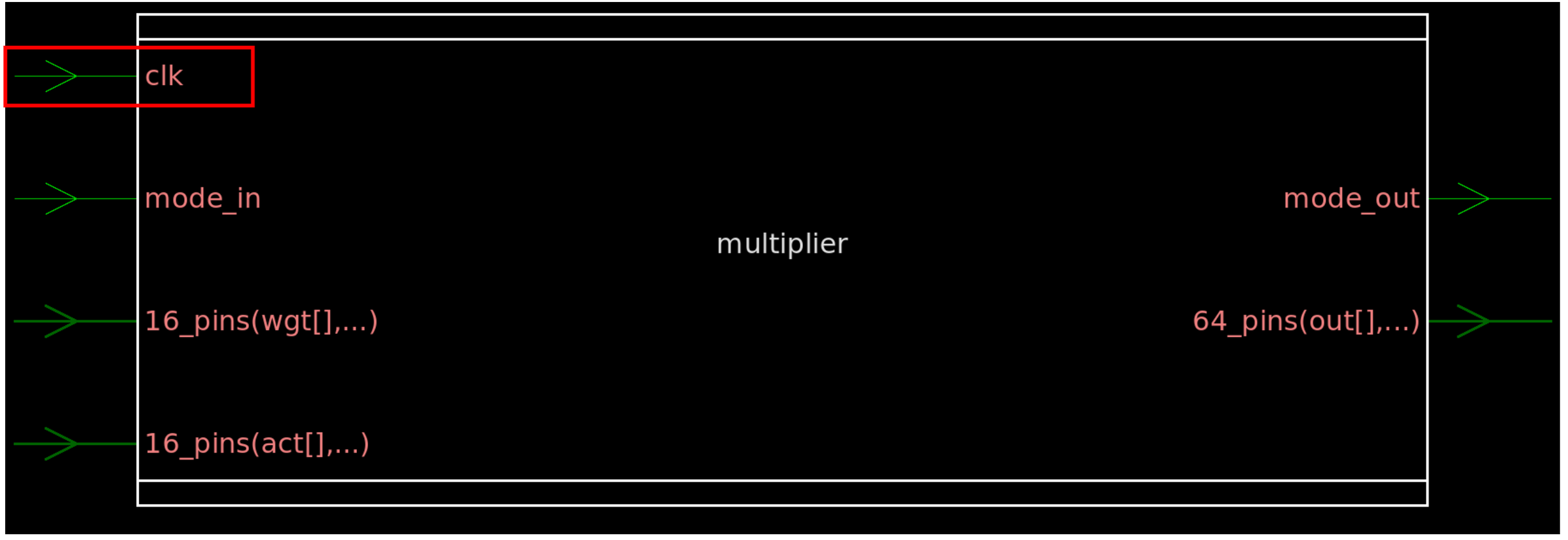
Number of Clock gating elements	4
Number of Gated registers	64 (16.98%)
Number of Ungated registers	313 (83.02%)
Total number of registers	377

Manual Clock Gating (4/4)

Design Compiler (TCL):

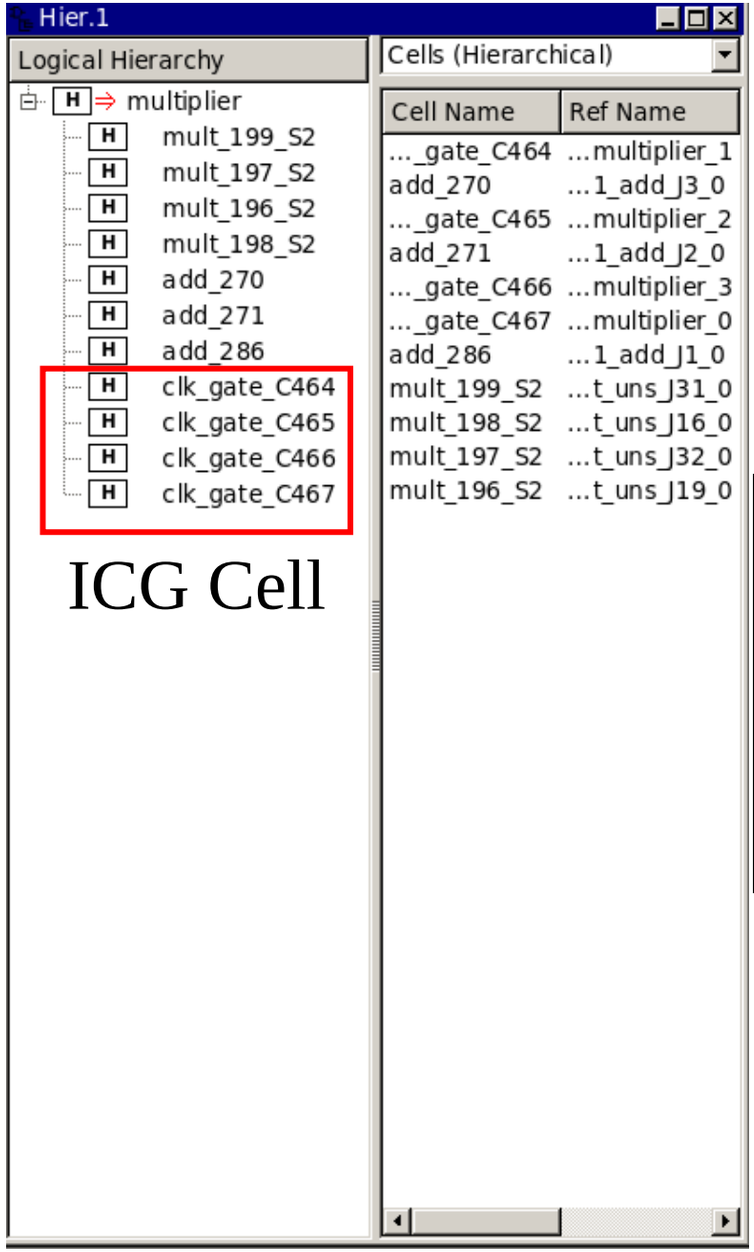
```
set_ideal_network [get_ports clk]
```

透過以上指令可將clock訊號線視為理想狀況 (可忽略High Fanout Synthesis之問題)。



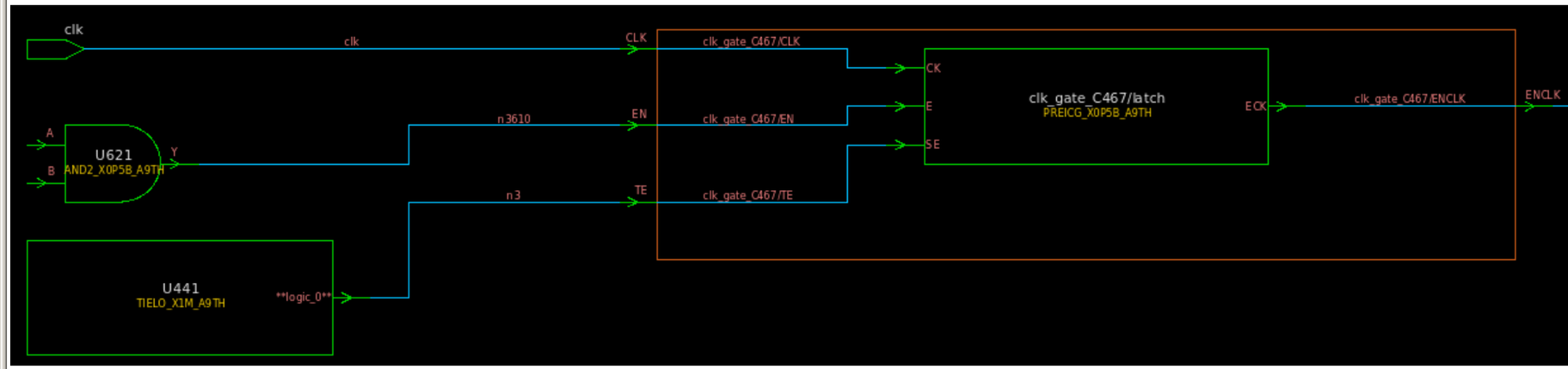
Manual Methods

GUI 介面顯示Clock Gating Cell (Example)



ICG Cell

ICG Cell



THANK YOU

感謝大家的聆聽