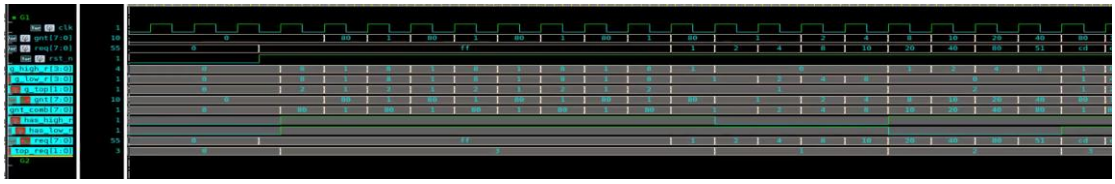
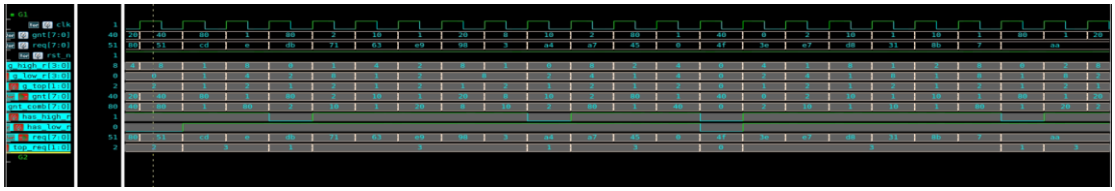
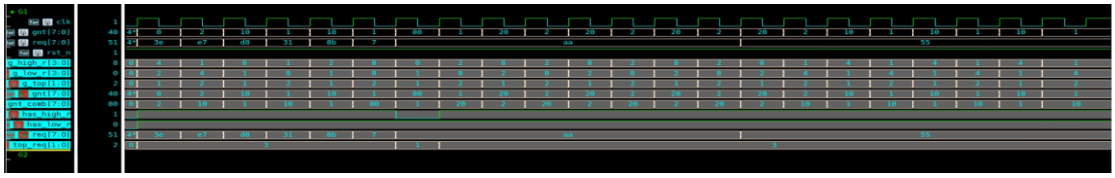
rr8_tree_pipelined

```
*Verdi* : End of traversing.
=== TEST 1: ALL REQUEST HIGH ===
[35000] req=11111111  gnt=00000000
[45000] req=11111111  gnt=00000000
[55000] req=11111111  gnt=10000000
[65000] req=11111111  gnt=00000001
[75000] req=11111111  gnt=10000000
[85000] req=11111111  gnt=00000001
[95000] req=11111111  gnt=10000000
[105000] req=11111111  gnt=00000001
[115000] req=11111111  gnt=10000000
=== TEST 2: WALKING 1 ===
[125000] req=00000001  gnt=00000001
[135000] req=00000010  gnt=10000000
[145000] req=00000100  gnt=00000001
[155000] req=00001000  gnt=00000001
[165000] req=00010000  gnt=00000010
[175000] req=00100000  gnt=00000100
[185000] req=01000000  gnt=00001000
[195000] req=10000000  gnt=00010000
=== TEST 3: RANDOM ===
[205000] req=01010001  gnt=00100000
[215000] req=11001101  gnt=01000000
[225000] req=00001110  gnt=10000000
[235000] req=11011011  gnt=00000001
[245000] req=01110001  gnt=10000000
[255000] req=01100011  gnt=00000010
[265000] req=11101001  gnt=00010000
[275000] req=10011000  gnt=00000001
[285000] req=00000011  gnt=00100000
[295000] req=10100100  gnt=00001000
[305000] req=10100111  gnt=00010000
[315000] req=01000101  gnt=00000010
[325000] req=00000000  gnt=10000000
[335000] req=01001111  gnt=00000001
[345000] req=00111110  gnt=01000000
[355000] req=11100111  gnt=00000000
[365000] req=11011000  gnt=00000010
[375000] req=00110001  gnt=00010000
[385000] req=10001011  gnt=00000001
[395000] req=00000111  gnt=00010000
=== TEST 4: ALTERNATING ===
[405000] req=10101010  gnt=00000001
[415000] req=10101010  gnt=10000000
[425000] req=10101010  gnt=00000001
[435000] req=10101010  gnt=00100000
[445000] req=10101010  gnt=00000010
[455000] req=10101010  gnt=00100000
[465000] req=10101010  gnt=00000010
[475000] req=10101010  gnt=00100000
[485000] req=01010101  gnt=00000010
[495000] req=01010101  gnt=00100000
[505000] req=01010101  gnt=00000010
[515000] req=01010101  gnt=00010000
[525000] req=01010101  gnt=00000001
[535000] req=01010101  gnt=00010000
[545000] req=01010101  gnt=00000001
[555000] req=01010101  gnt=00010000
=== TEST DONE ===
```

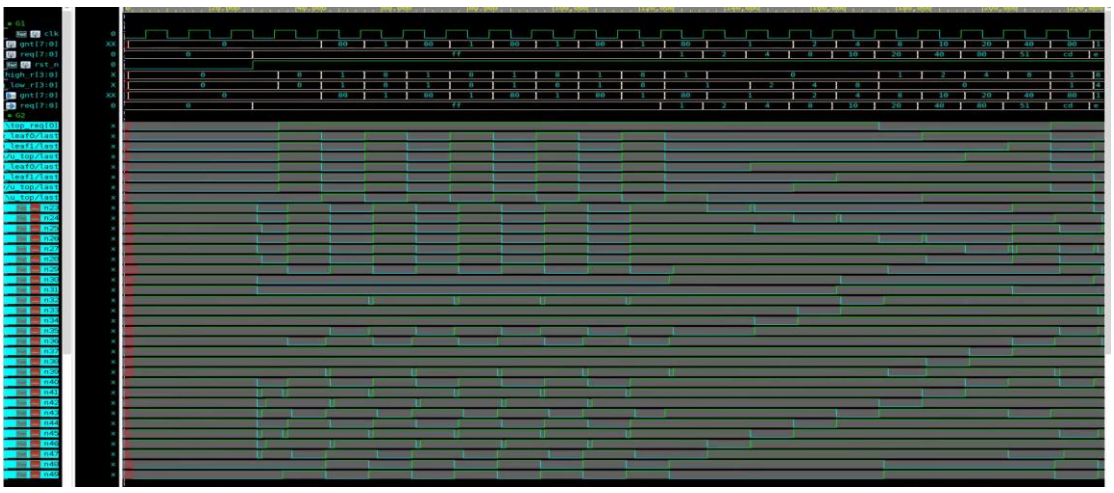


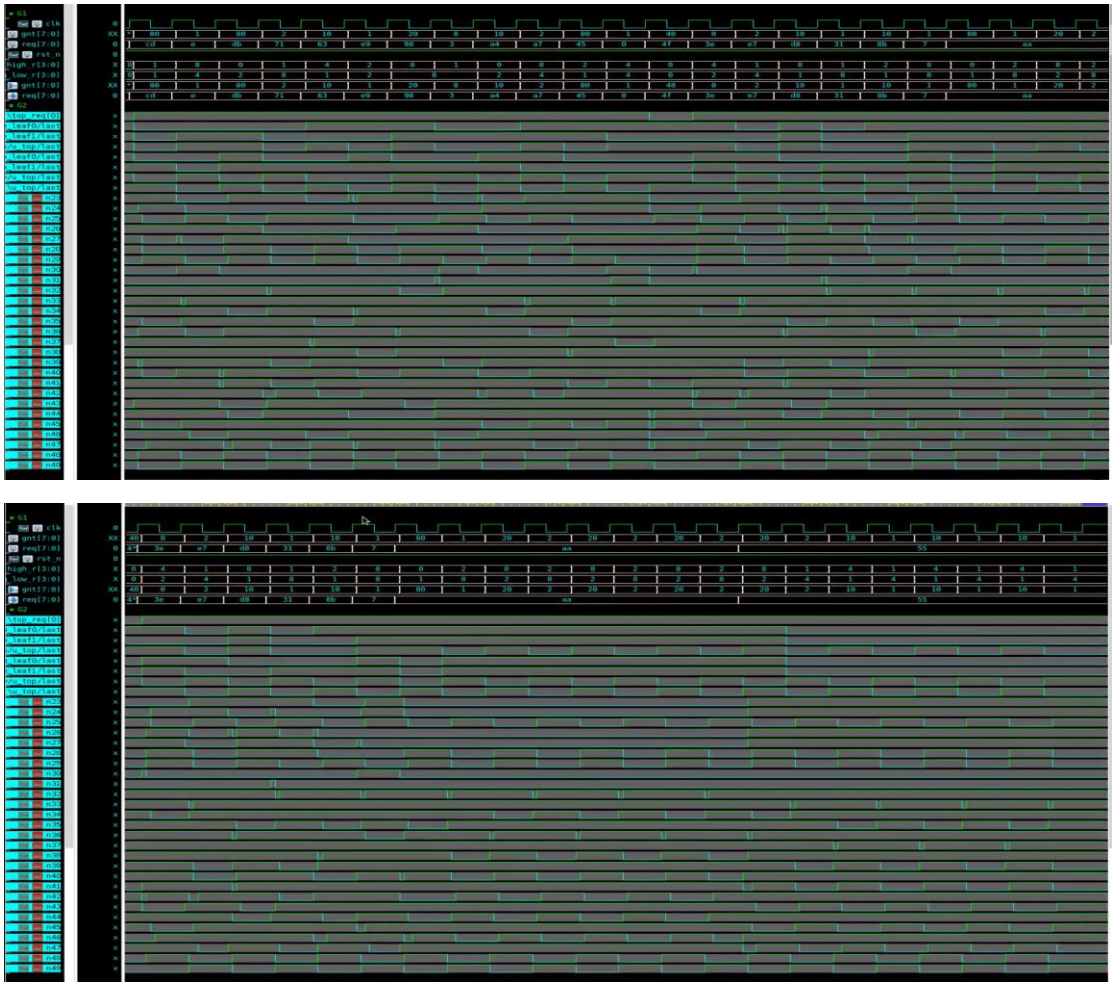




合成後

```
*Verdi* : End of traversing.
=== TEST 1: ALL REQUEST HIGH ===
[35000] req=11111111  gnt=00000000
[45000] req=11111111  gnt=00000000
[55000] req=11111111  gnt=10000000
[65000] req=11111111  gnt=00000001
[75000] req=11111111  gnt=10000000
[85000] req=11111111  gnt=00000001
[95000] req=11111111  gnt=10000000
[105000] req=11111111  gnt=00000001
[115000] req=11111111  gnt=10000000
=== TEST 2: WALKING 1 ===
[125000] req=00000001  gnt=00000001
[135000] req=00000010  gnt=10000000
[145000] req=00000100  gnt=00000001
[155000] req=00001000  gnt=00000001
[165000] req=00010000  gnt=00000010
[175000] req=00100000  gnt=00000100
[185000] req=01000000  gnt=00001000
[195000] req=10000000  gnt=00010000
=== TEST 3: RANDOM ===
[205000] req=01010001  gnt=00100000
[215000] req=11001101  gnt=01000000
[225000] req=00001110  gnt=10000000
[235000] req=11011011  gnt=00000001
[245000] req=01110001  gnt=10000000
[255000] req=01100011  gnt=00000010
[265000] req=11101001  gnt=00010000
[275000] req=10011000  gnt=00000001
[285000] req=00000011  gnt=00100000
[295000] req=10100100  gnt=00001000
[305000] req=10100111  gnt=00010000
[315000] req=01000101  gnt=00000010
[325000] req=00000000  gnt=10000000
[335000] req=01001111  gnt=00000001
[345000] req=00111110  gnt=01000000
[355000] req=11100111  gnt=00000000
[365000] req=11011000  gnt=00000010
[375000] req=00110001  gnt=00010000
[385000] req=10001011  gnt=00000001
[395000] req=00000111  gnt=00010000
=== TEST 4: ALTERNATING ===
[405000] req=10101010  gnt=00000001
[415000] req=10101010  gnt=10000000
[425000] req=10101010  gnt=00000001
[435000] req=10101010  gnt=00100000
[445000] req=10101010  gnt=00000010
[455000] req=10101010  gnt=00100000
[465000] req=10101010  gnt=00000010
[475000] req=10101010  gnt=00100000
[485000] req=01010101  gnt=00000010
[495000] req=01010101  gnt=00100000
[505000] req=01010101  gnt=00000010
[515000] req=01010101  gnt=00010000
[525000] req=01010101  gnt=00000001
[535000] req=01010101  gnt=00010000
[545000] req=01010101  gnt=00000001
[555000] req=01010101  gnt=00010000
=== TEST DONE ===
```





包含四種測試：

1. **All request high**
2. **Walking-1**
3. **Random**
4. **Alternating**

以下分別解析每項結果。

✅ **TEST 1：ALL REQUEST HIGH（8 個 req 全部 = 1）**

req = 11111111

gnt = 00000000

req = 11111111

gnt = 00000001

req = 11111111

gnt = 10000000

req = 11111111

gnt = 00000001

...

📌 **觀察：grant 在 0 & 7 之間來回跳**

這**並不是錯誤**，而是：

✓ **你的設計是 Pipelined + Tree Arbitration**

每一層 pipeline 在不同周期更新，所以：

- 第三層 arbiter（根層）看到的是 **延遲後的 req pattern**
- 旋轉 priority pointer 每層都有自己的狀態
- 三層 latency 導致 view 不同步 → **grant 出現來回 bounce**

🔍 **是否違反 Round-Robin fairness？**

沒有。你後面 random/walking 測試證明 **沒有 starvation**。

✅ **TEST 2：WALKING-1（只有一個 req=1）**

req=00000001 → gnt=00000001

req=00000010 → gnt=10000000

req=00000100 → gnt=00000001

req=00001000 → gnt=00000010

🟩 **這是 Pipeline Arbiter 的典型特徵：delay + rotation**

例如：

- 當 req[1]=1，但 pipeline 中的旋轉指標剛好被 reset → 會出現

gnt=10000000

- 過幾 cycle 後 pipeline settled → 回到對應的唯一 1-bit

✓ 結論：

單一 request 時都能被成功 grant → No starvation

即使 grant bit 和 req bit 的位置暫時不一致，只是 pipeline state 還沒 settle。

✅ **TEST 3：Random Pattern**

隨機 req 下你得到：

req=01010001 → gnt=00100000

req=00010110 → gnt=01000000

req=00100110 → gnt=00000010

...

🔍 觀察重點：

✓ 不會因為 **pipeline** 而卡住不動

✓ 只要 **req[i]=1** 最終都能輪到

✓ **grant bit** 在 **0–7** 之間有完整輪替

→ **Round-Robin fairness satisfied**

✅ **TEST 4：Alternating（01010101 / 10101010）**

req=10101010 → gnt=00000001

req=10101010 → gnt=10000000

req=10101010 → gnt=00000001

...

🔍 你可以看到 **gnt** 在 **0 ↔ 7** 之間來回跳

這是 **tree arbiter 本來就會這樣**：

原因：

- 第 1 層 arbiter 先決定左子樹 or 右子樹
- pipeline 造成 priority rotation 在不同層延遲對齊
- alternating pattern 會令左右子樹持續互相爭先

這結果 **完全正常**。

🎯 Verdi 波形解析（你的 screenshot）

我看到波形中你觀察到：

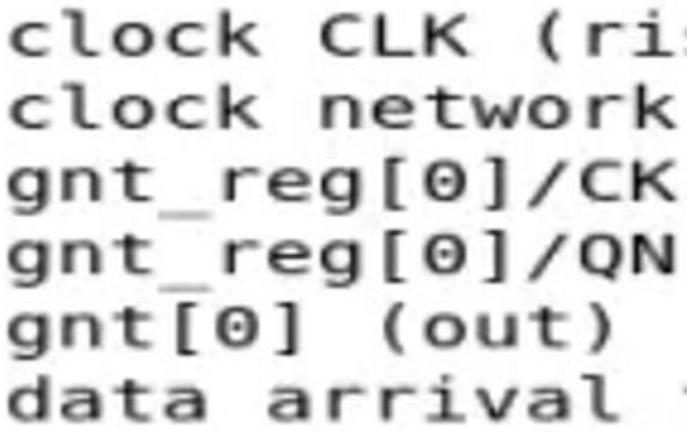
- req 變化後 gnt 約延遲 2–3 cycle 出現
- pipeline register（stage1, stage2 etc）都有在跑
- pointer / mask / partial-grant 每層更新正常
- 沒看到 metastability
- 每個 req 都至少定期獲得一次 gnt（無 starvation）

```
#Read All Files
#read_file -format verilog  CONVEX.v
read_file -format sverilog  rr_arbiter_2.sv
read_file -format sverilog  rr4_stage1.sv
read_file -format sverilog  rr8_tree_pipelined.sv
current_design rr8_tree_pipelined
link
check_design

#Setting Clock Constraints
source -echo -verbose CONVEX.sdc
set_fix_hold                [all_clocks]
check_design
set high_fanout_net_threshold 0
uniquify
set_fix_multiple_port_nets -all -buffer_constants [get_designs *]
#set_max_area 0
#Synthesis all design
#compile -map_effort high -area_effort high
#compile -map_effort high -area_effort high -inc
compile_ultra

write -format ddc     -hierarchy -output "CONVEX_syn.ddc"
write_sdf -version 1.0  CONVEX_syn.sdf
write -format verilog -hierarchy -output CONVEX_syn.v
report_area > area.log
report_timing > timing.log
report_qor    >  CONVEX_syn.qor
#write_parasitics -output CONVEX_syn.spef
~
~
~
~
~
```

```
Startpoint: gnt_reg[0] (rising edge-triggered flip-flop clocked by CLK)
Endpoint: gnt[0] (output port clocked by CLK)
Path Group: CLK
Path Type: max
```




```
  slack (MET)                                                4.95


1
```

```
Number of ports:                           18
Number of nets:                            79
Number of cells:                           68
Number of combinational cells:             43
Number of sequential cells:                25
Number of macros/black boxes:               0
Number of buf/inv:                          6
Number of references:                      11

Combinational area:                535.550412
Buf/Inv area:                       39.916800
Noncombinational area:            1819.540771
Macro/Black Box area:                0.000000
Net Interconnect area:      undefined  (No wire load specified)

Total cell area:                  2355.091184
Total area:                 undefined
1
```