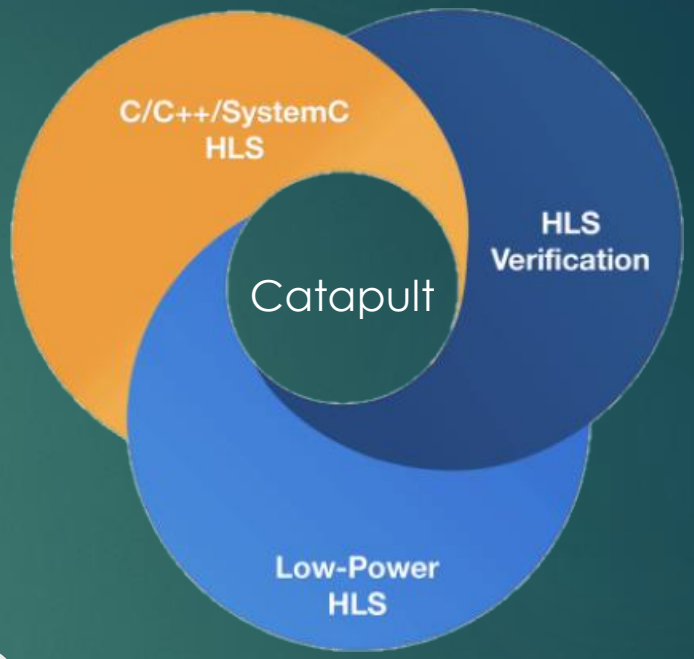
HLS DUTh Lab

ΣΑΜΟΛΑΔΑΣ ΤΡΙΑΝΤΑΦΥΛΛΟΣ 57259



HLS DUTh Lab

Code used:

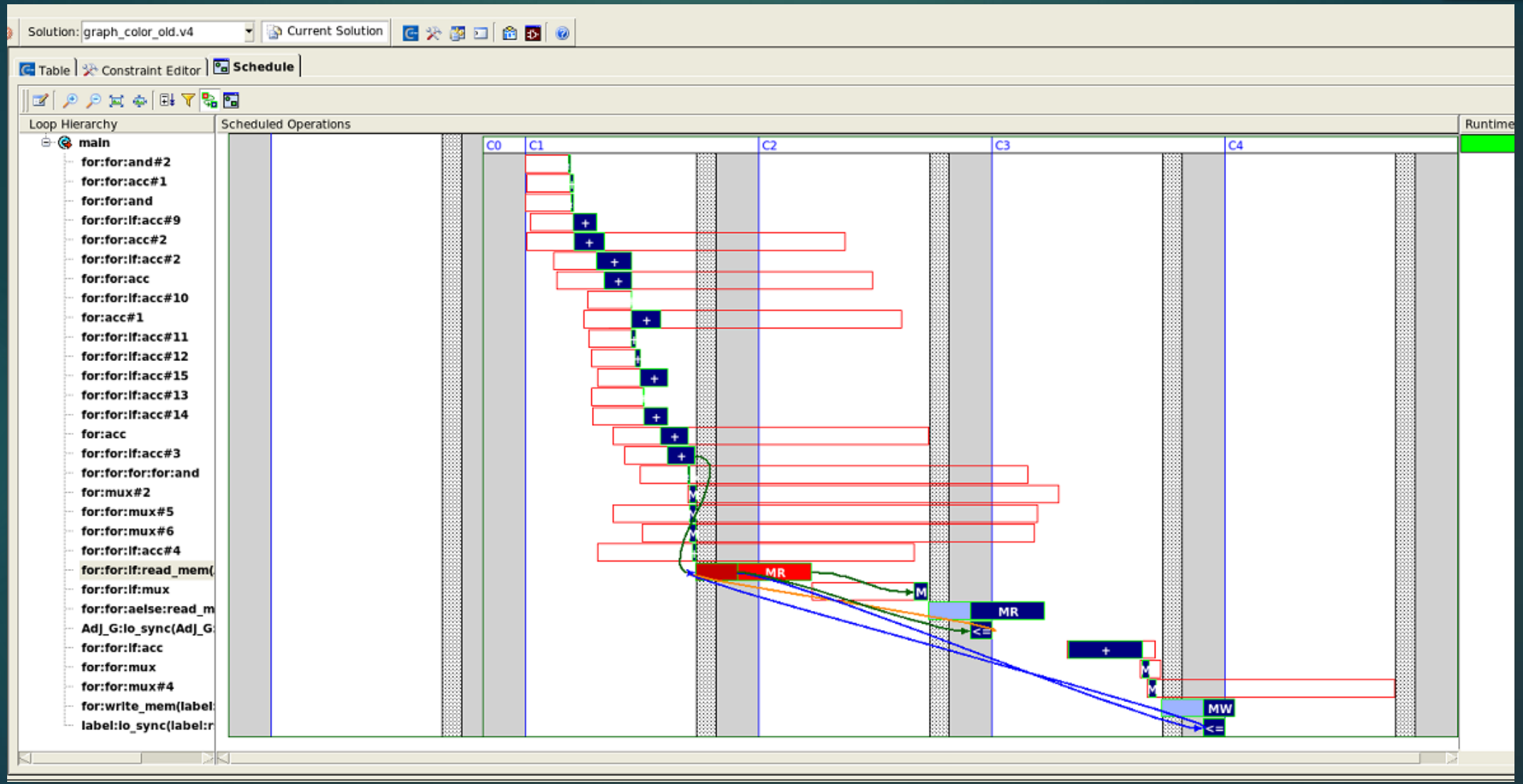
```
#pragma hls_design top
int CCS_BLOCK(graph_color_old)(ac_int<N,false> Adj_G[N],int label[N]){
    for (int i=0;i<N;i++){
        int c = 1;
        for (int j= 0; j<N; j++){
            if( Adj_G[i][5-j] == 1 && label[j] == c){ //if you are connected to this node and has same color
                c += 1;   // pick next color
            }
        }
        label[i] = c;
    }
    CCS_RETURN(label[N]);
}
```

HLS DUTh Lab

Solution	Latency...	Latency...	Throughput Cycles	Throughput Time	Total Area	Slack
graph_color_old.v1 (extract)	82	164.00	86	172.00	1291.33	0.10
graph_color_old.v2 (extract)	118	236.00	122	244.00	763.47	0.52
graph_color_old.v3 (allocate)						
graph_color_old.v4 (extract)	**72**	**144.00**	**72**	**144.00**	**798.64**	**0.62**
graph_color_old.v5 (allocate)	72	144.00	72	144.00	964.43	
graph_color_old.v6 (allocate)						
graph_color_old.v7 (allocate)	144	288.00	144	288.00	1876.26	
graph_color_old.v8 (allocate)						
graph_color_old.v9 (extract)	71	142.00	72	144.00	1284.10	0.51

- .v1 -> Raw Code
- .v2 -> Adjacency Matrix with Singleport and *label[N]* with Dualport Memory Interface
- **.v4 -> Apply(at .v2) Loop Pipeline with II = 2**
- .v5 -> *label[N]* with Singleport memory

HLS DUTh Lab

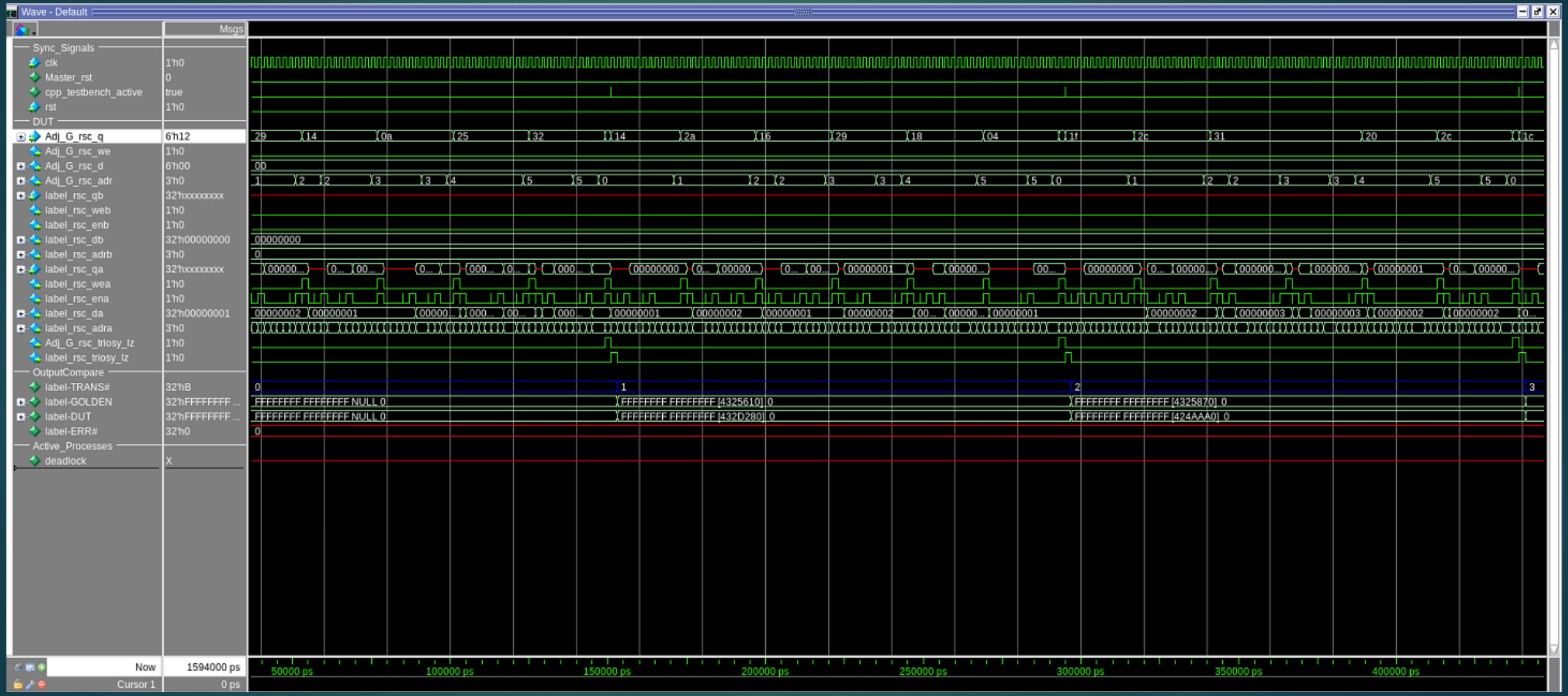

Solution: graph_color_old.v4
Current Solution
Table
Constraint Editor
Schedule
Loop Hierarchy
Scheduled Operations
Runtime
main
for:for:and#2
for:for:acc#1
for:for:and
for:for:lf:acc#9
for:for:acc#2
for:for:lf:acc#2
for:for:acc
for:for:lf:acc#10
for:acc#1
for:for:lf:acc#11
for:for:lf:acc#12
for:for:lf:acc#15
for:for:lf:acc#13
for:for:lf:acc#14
for:acc
for:for:lf:acc#3
for:for:for:for:and
for:mux#2
for:for:mux#5
for:for:mux#6
for:for:lf:acc#4
for:for:lf:read_mem(
for:for:lf:mux
for:for:aelse:read_m
Adj_G:lo_sync(Adj_G
for:for:lf:acc
for:for:mux
for:for:mux#4
for:write_mem(label
label:lo_sync(label:r
C0
C1
C2
C3
C4
MR
MR
MW

HLS DUTh Lab

```
# Chromatic Number: 4
# ----------------------------------------
# Generate random graph
# 0 0 0 0 0 0
# 0 0 0 1 1 0
# 0 0 0 1 0 0
# 0 1 1 0 0 1
# 0 1 0 0 0 0
# 0 0 0 1 0 0
# Chromatic Number: 2
# ----------------------------------------
# Generate random graph
# 0 0 1 0 0 0
# 0 0 0 1 1 1
# 1 0 0 1 1 0
# 0 1 1 0 0 0
# 0 1 1 0 0 0
# 0 1 0 0 0 0
# Chromatic Number: 3
# ----------------------------------------
# Info: Execution of user-supplied C++ testbench 'main()' has completed with exit code = 0
#
# Info: Collecting data completed
#    captured 11 values of Adj_G
#    captured 11 values of label_IN
#    captured 11 values of label
#    captured 0 values of return
# Info: scverify_top/user_tb: Simulation completed
#
# Checking results
# 'label'
#    capture count       = 11
#    comparison count    = 11
#    ignore count        = 0
#    error count         = 0
#    stuck in dut fifo   = 0
#    stuck in golden fifo = 0
# 'return' - warning, output was optimized away
#
# Info: scverify_top/user_tb: Simulation PASSED @ 1594 ns
# ** Note: (vsim-6574) SystemC simulation stopped by user.
# 1
#
```

Simulation Passed : 1594ns

HLS DUTh Lab


Wave - Default
Msgs
Sync_Signals
clk
Master_rst
cpp_testbench_active
rst
DUT
Adj_G_rsc_q
Adj_G_rsc_we
Adj_G_rsc_d
Adj_G_rsc_adr
label_rsc_qb
label_rsc_web
label_rsc_enb
label_rsc_db
label_rsc_adrb
label_rsc_qa
label_rsc_wea
label_rsc_ena
label_rsc_da
label_rsc_adra
Adj_G_rsc_triosy_lz
label_rsc_triosy_lz
OutputCompare
label-TRANS#
label-GOLDEN
label-DUT
label-ERR#
Active_Processes
deadlock
Now
1594000 ps
Cursor 1
0 ps
50000 ps
100000 ps
150000 ps
200000 ps
250000 ps
300000 ps
350000 ps
400000 ps

HLS DUTh Lab

.v12 Use temp variable to store the line of Adj_G[i]. So one access for each iteration of the outer loop. (We spare accessing the memory at the if-statement)