



Rensselaer

why not change the world?®

Embedded Neural Network

Ashton Ropp

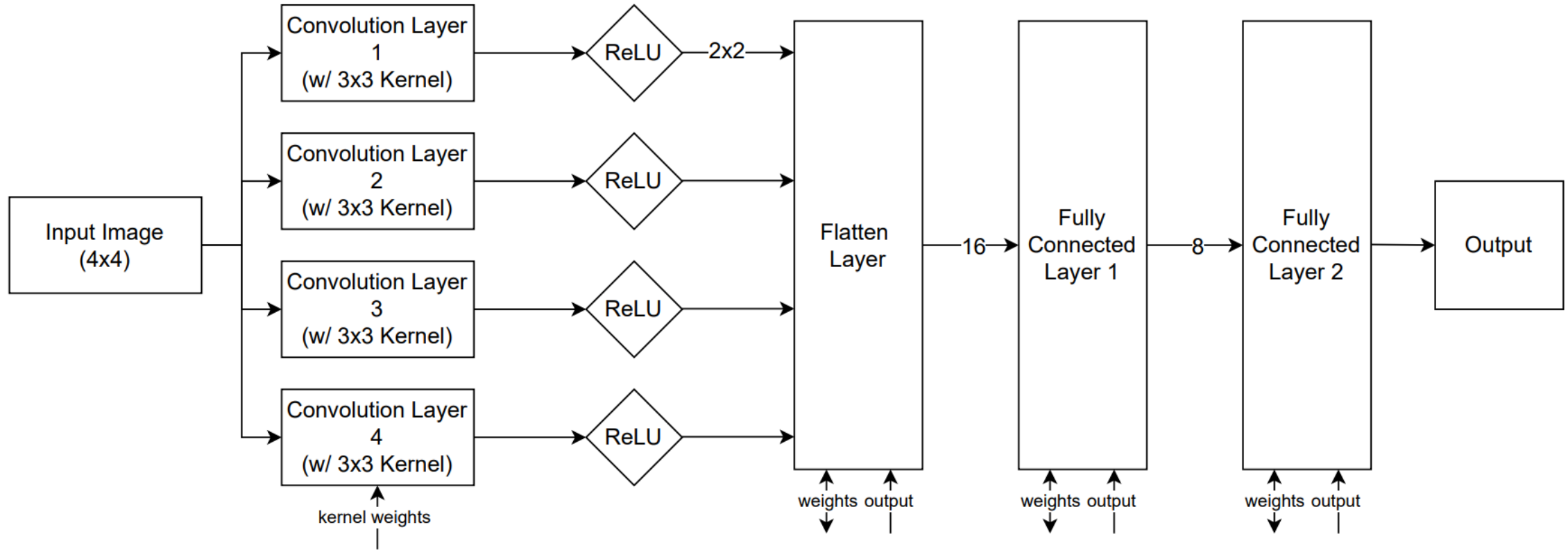
4/17/25

Most Basic Neural Network Overview

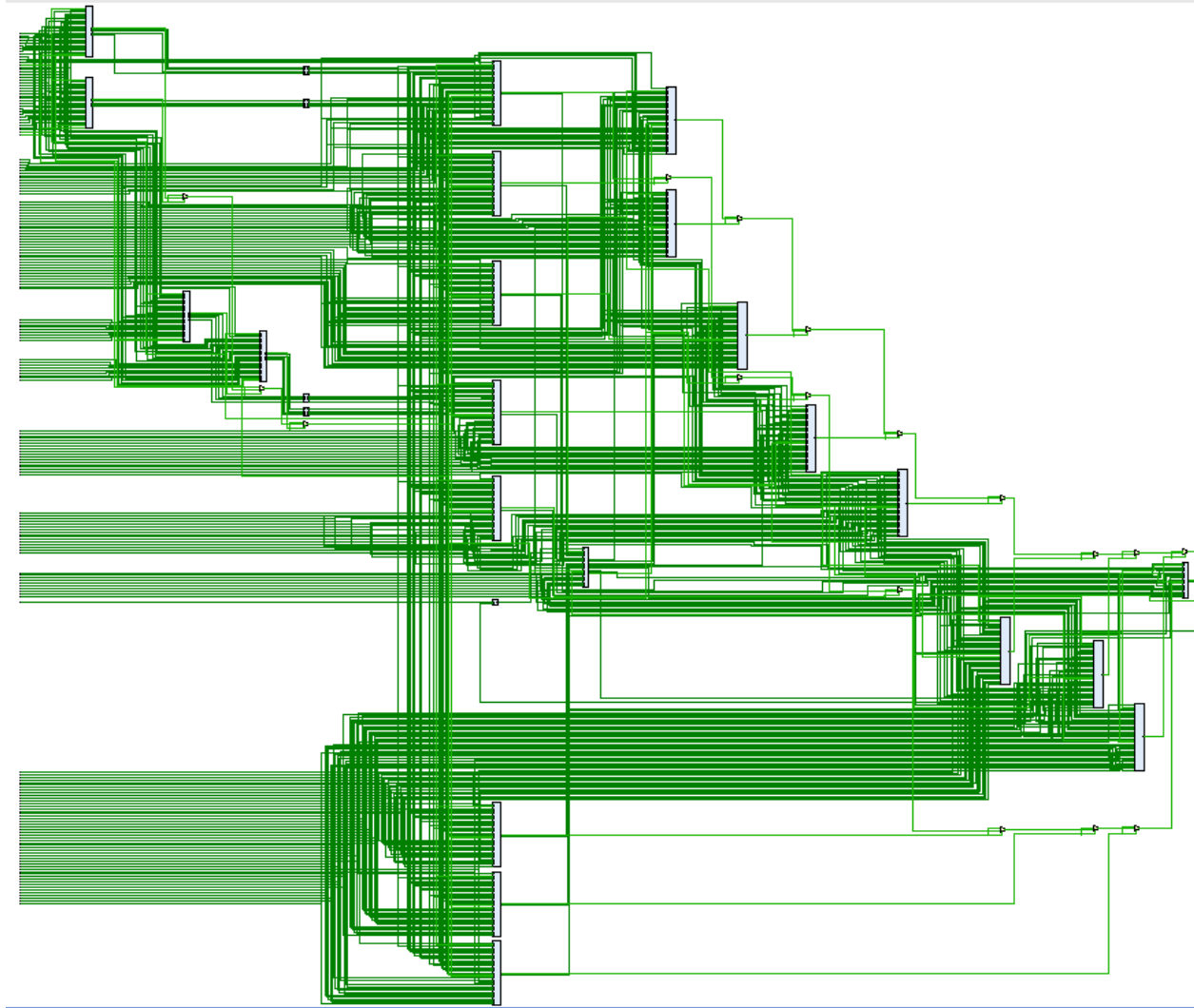
- Prediction: is the 4x4 image all black or all white (trivial)
 - Easy to extend
- Four convolutional filters (3x3)
- ReLU activation
- Weights stored externally: system only responsible for evaluation and backpropagation

Signal Name	Type	Description
input_image[4][4]	logic signed [15:0]	4×4 image of pixel values in Q8.8 fixed point
conv_weights[4][3][3]	logic signed [15:0]	4 convolution filters (each 3×3), for feature extraction
fc1_weights[8][16]	logic signed [15:0]	FC1 weights, 8 neurons × 16 flattened inputs
fc1_bias[8]	logic signed [15:0]	Biases for each of the 8 neurons in FC1
fc2_weights[8]	logic signed [15:0]	Final layer weights (maps 8 FC1 outputs to 1 output)
fc2_bias	logic signed [15:0]	Single bias value for FC2
label	logic signed [15:0]	Ground truth label (supervised training target)
learning_rate	logic signed [15:0]	Used to scale gradient updates
clk, rst, start	logic	Control signals

System Structure

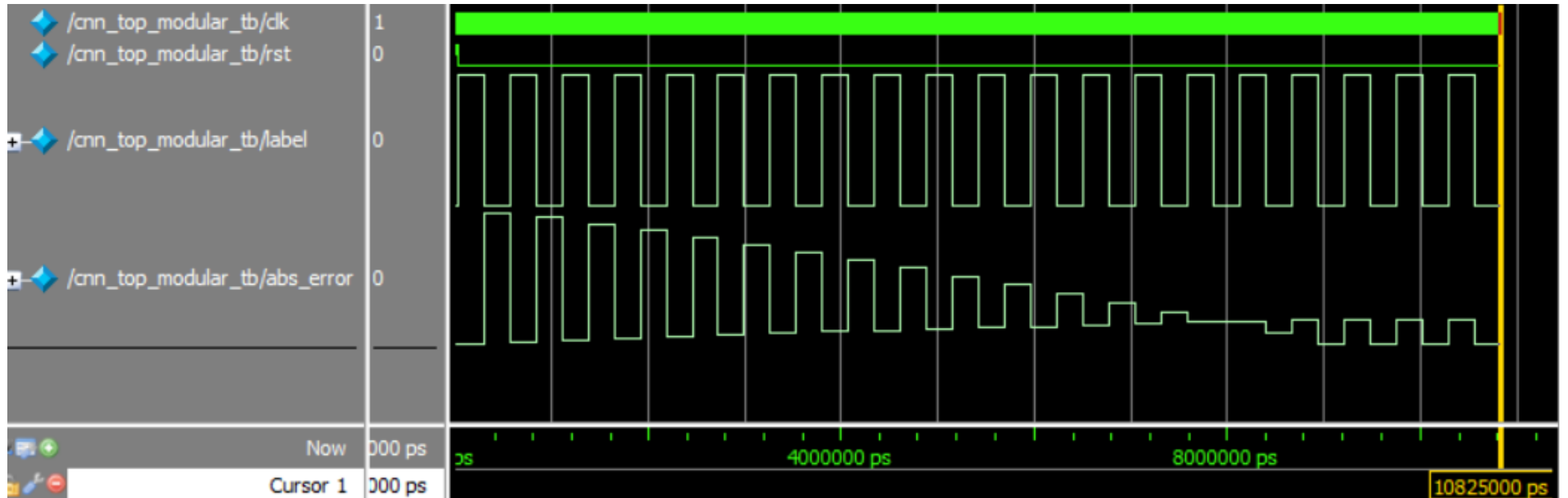


Vivado Schematic



```
# === Epoch 14 ===  
# Prediction: 220 (~= 0.86) | Label: 256 | Error: 35  
# Prediction: 37 (~= 0.14) | Label: 0 | Error: 36  
#  
# === Epoch 15 ===  
# Prediction: 238 (~= 0.93) | Label: 256 | Error: 37  
# Prediction: 40 (~= 0.16) | Label: 0 | Error: 18  
#  
# === Epoch 16 ===  
# Prediction: 256 (~= 1.00) | Label: 256 | Error: 40  
# Prediction: 40 (~= 0.16) | Label: 0 | Error: 0  
#  
# === Epoch 17 ===  
# Prediction: 256 (~= 1.00) | Label: 256 | Error: 40  
# Prediction: 40 (~= 0.16) | Label: 0 | Error: 0  
#  
# === Epoch 18 ===  
# Prediction: 256 (~= 1.00) | Label: 256 | Error: 40  
# Prediction: 40 (~= 0.16) | Label: 0 | Error: 0  
#  
# === Epoch 19 ===  
# Prediction: 256 (~= 1.00) | Label: 256 | Error: 40  
# Prediction: 40 (~= 0.16) | Label: 0 | Error: 0
```

ModelSim



Optimizations Part A

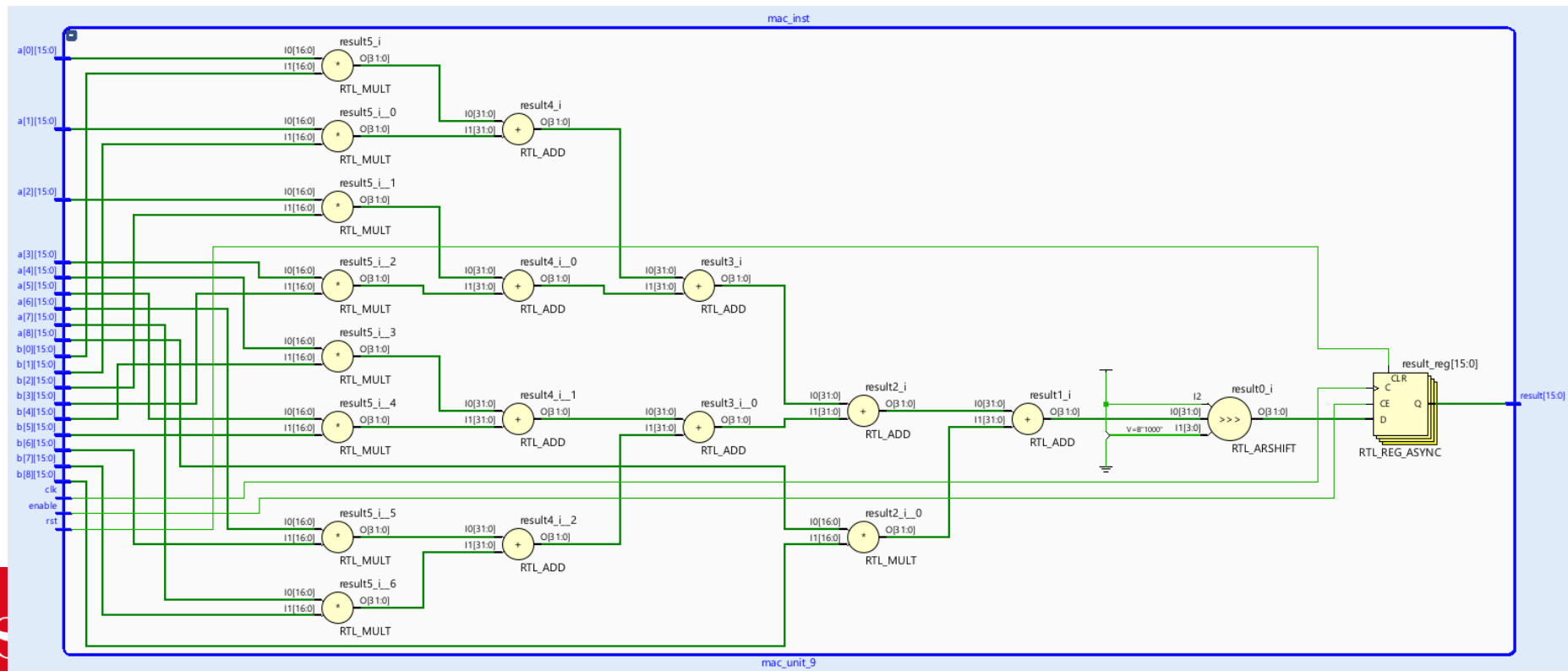
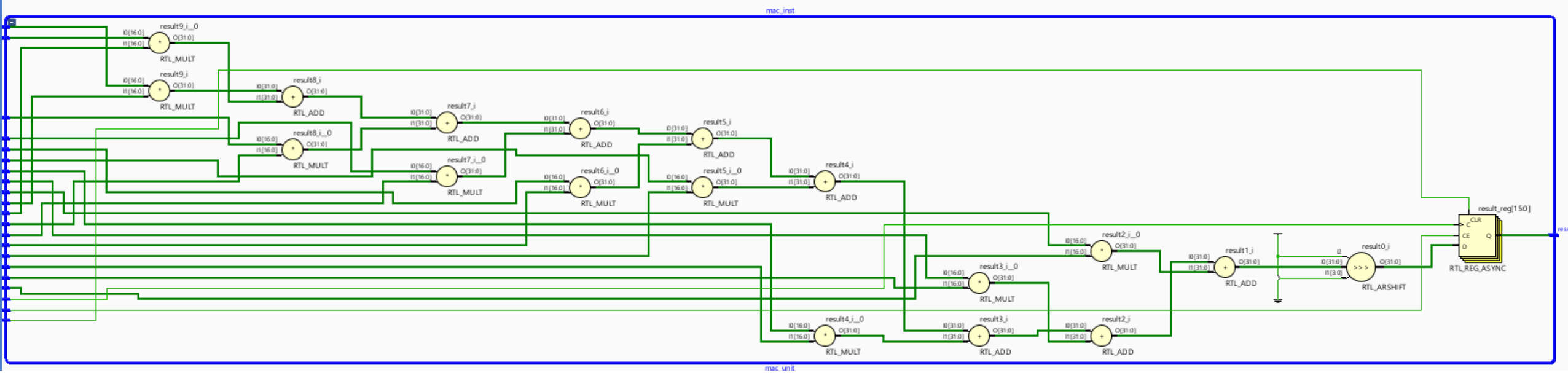
- Convolution unit: shown below
 - Explicit unrolling avoids runtime logic for index
- MAC: adder tree (next slide)

```
int k;  
always_comb begin  
    k = 0;  
    for (int m = 0; m < KERNEL_SIZE; m++) begin  
        for (int n = 0; n < KERNEL_SIZE; n++) begin  
            b_flat[k] = kernel_weights[m][n];  
            k++;  
        end  
    end  
end
```

Old

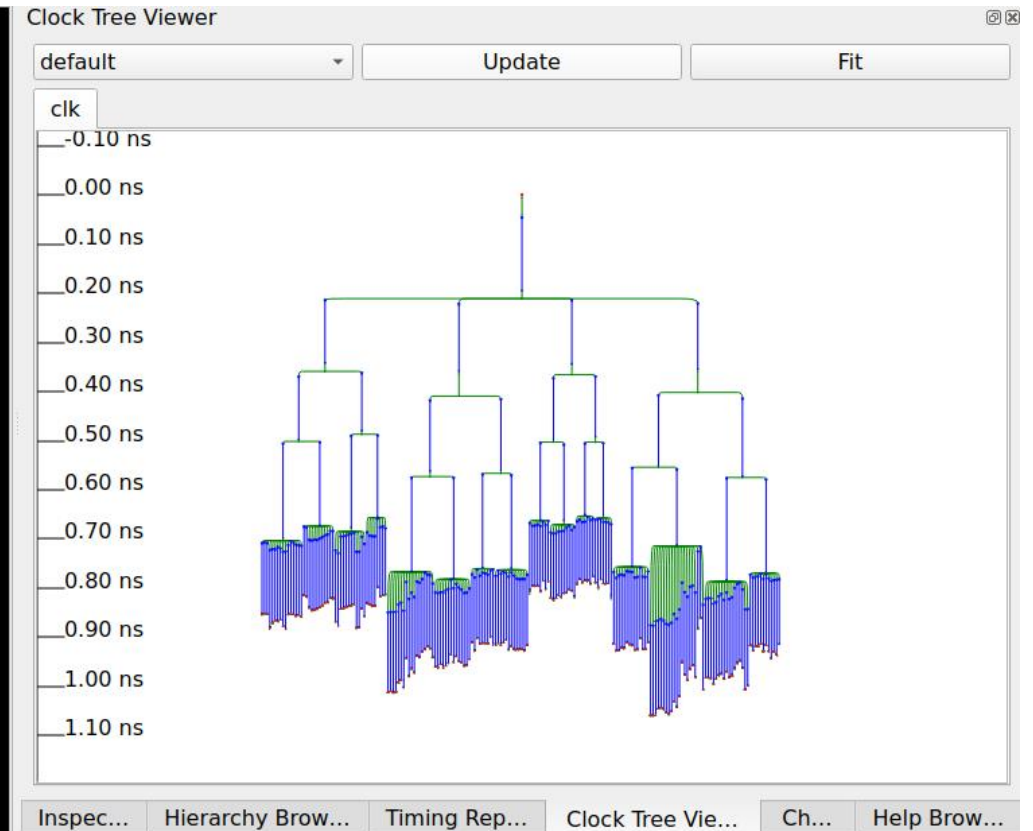
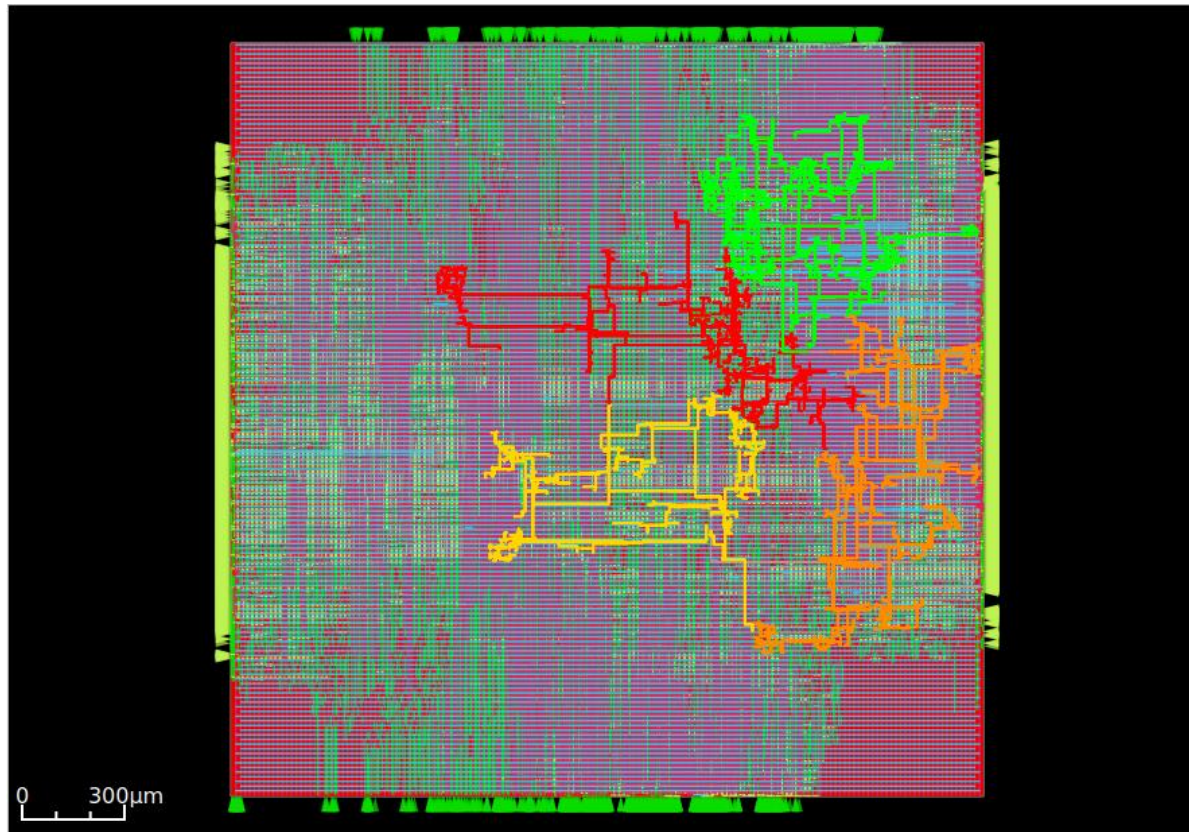
```
genvar m, n;  
generate  
    for (m = 0; m < KERNEL_SIZE; m++) begin : FLATTEN_M  
        for (n = 0; n < KERNEL_SIZE; n++) begin : FLATTEN_N  
            localparam int IDX = m * KERNEL_SIZE + n;  
            assign b_flat[IDX] = kernel_weights[m][n];  
        end  
    end  
endgenerate
```

New



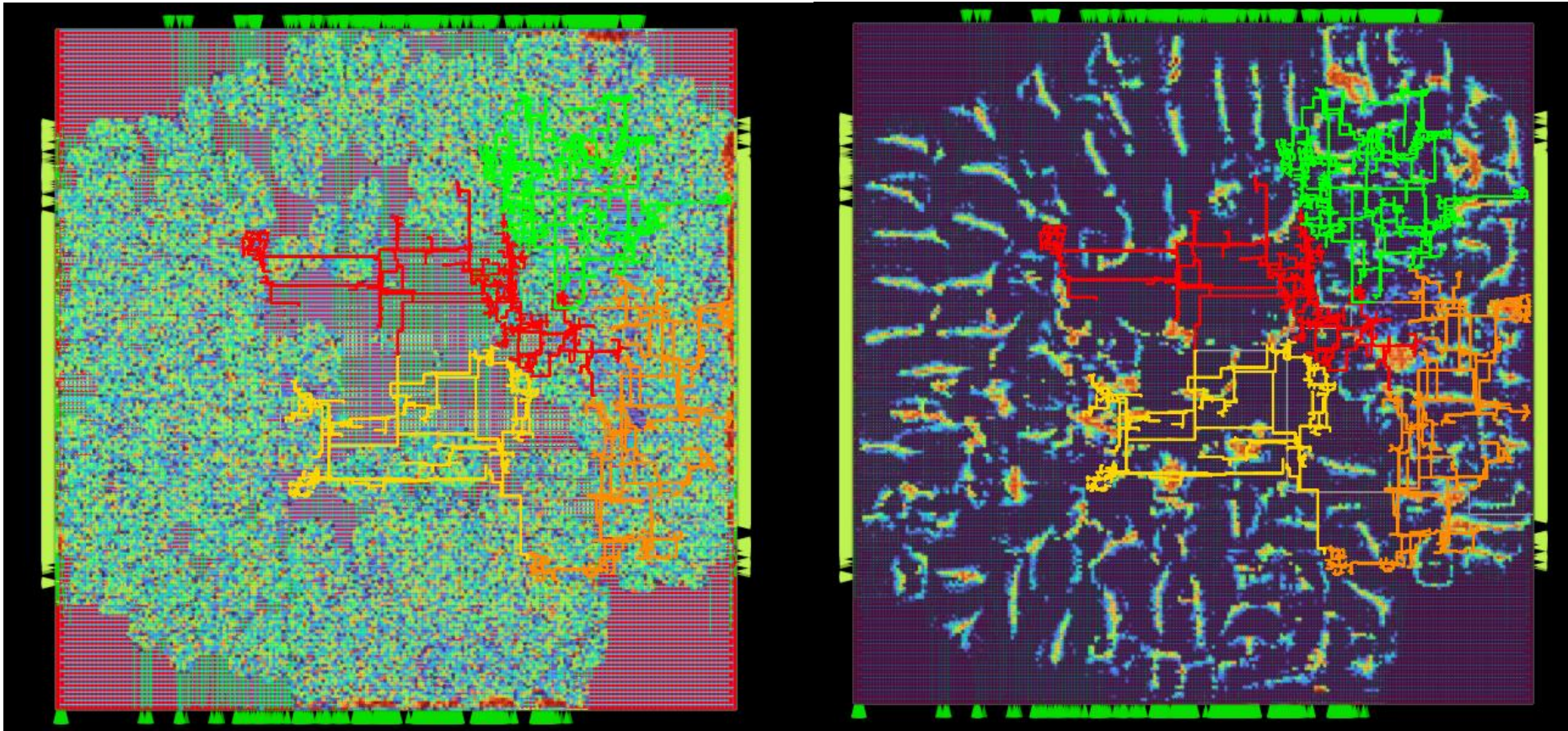
Results - Optimized Part A

- Full chip and clock tree

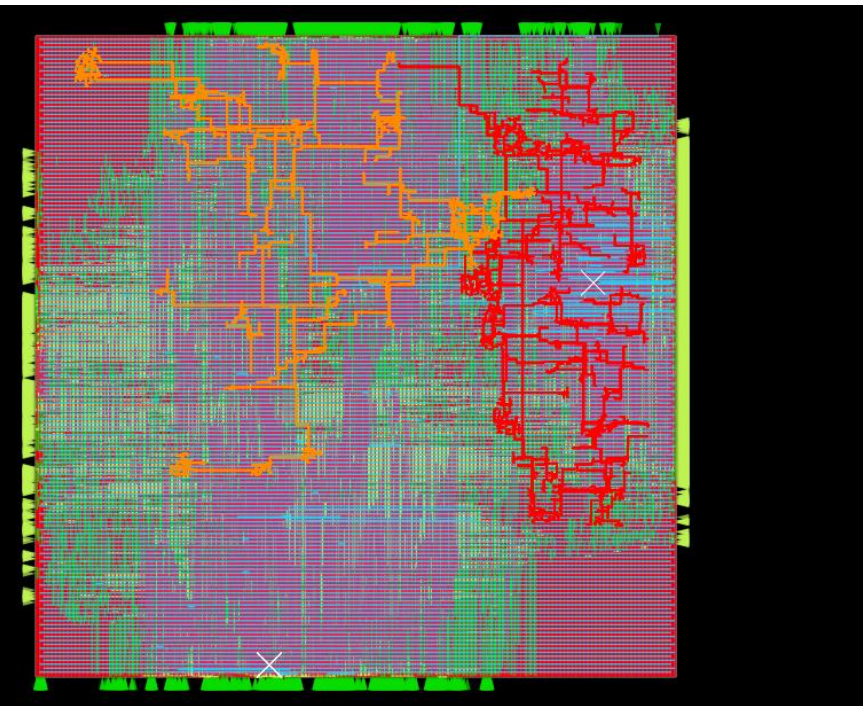


Results - Optimized Part A (Misc)

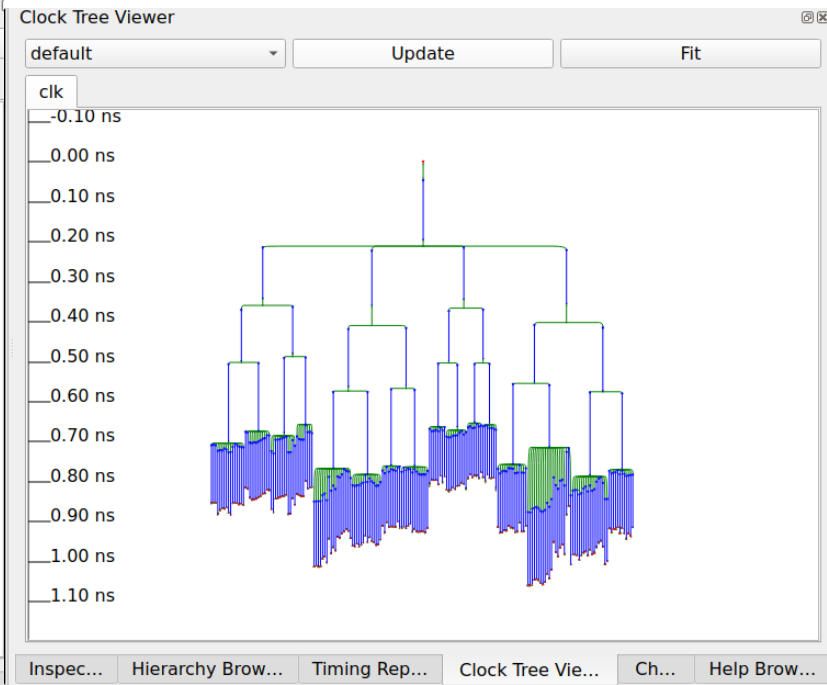
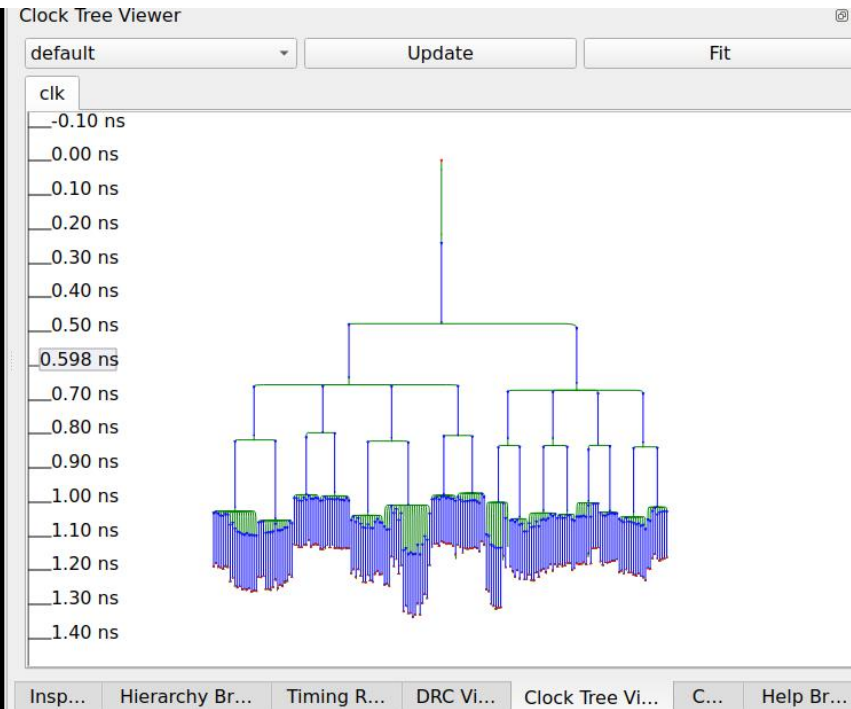
- Placement density and power density



Results - Comparison



Unoptimized



Optimized

Part B Optimizations

- Loop unrolling: 16 cycle delay to 1 cycle delay
- Around a dozen of these exist on the chip

```
COMPUTE: begin
  weights_out[idx] <= weights_in[idx] - update_mul[idx][23:8];
  dL_drelu[idx]    <= backprop_mul[idx][23:8];

  if (idx == INPUT_DIM - 1) begin
    bias_out <= bias_in - bias_update[23:8];
    state <= DONE;
  end else begin
    idx <= idx + 1;
  end
end
```

Old

```
COMPUTE: begin
  for (int i = 0; i < INPUT_DIM; i++) begin
    weights_out[i] <= weights_in[i] - update_mul[i][23:8];
    dL_drelu[i]    <= backprop_mul[i][23:8];
  end
  bias_out <= bias_in - bias_update[23:8];
  state    <= DONE;
end
```

New

Further Optimizations

- Added parallel MAC units to compute all 3x3 filters on 4x4 image in one cycle (trade area, power for speed)
- Removed FSM from as many layers as possible
- Kept all previous optimizations

Results

- 10825 ns to 6000 ns to 2000 ns for full test
- 5 cycles for entire pipeline and backpropagation
- A: 4.5% area decrease, 35% power decrease, 44% time decrease
- B: 47% area increase, 10% power increase, 82% time decrease

Design	Area (um ²)	WNS	TNS	Power (W)	Time (ns)
Unoptimized	2370830	-2.52	-136.61	11.10	10825
Optimized A	2265345	0.17	0	7.22	6000
Optimized B	3489451	-1.07	-68.37	12.28	2000

OpenROAD Metrics

Future Work

- Figure out why power numbers are exponentially large
- Close timing issues
- More clock and PPA tuning
- KLayout
- Synopsis Design Compiler

Synopsis Design Compiler Results (Part A Optimizations)

Number of ports:	86765
Number of nets:	550240
Number of cells:	398389
Number of combinational cells:	391140
Number of sequential cells:	6050
Number of macros/black boxes:	0
Number of buf/inv:	110400
Number of references:	30
Combinational area:	1217524.000000
Buf/Inv area:	110400.000000
Noncombinational area:	43506.000000
Macro/Black Box area:	0.000000
Net Interconnect area:	undefined (No wire load specified)
Total cell area:	1261030.000000

Unoptimized

Number of ports:	86765
Number of nets:	595693
Number of cells:	438107
Number of combinational cells:	430814
Number of sequential cells:	6050
Number of macros/black boxes:	0
Number of buf/inv:	50329
Number of references:	41
Combinational area:	1426731.000000
Buf/Inv area:	63735.000000
Noncombinational area:	43897.000000
Macro/Black Box area:	0.000000
Net Interconnect area:	undefined (No wire load specified)
Total cell area:	1470628.000000

Part A