



Integration of a datapath generation with an ASTRO flow

Kossay Omary & Sunny Shin,
C2 Microsystems

Fuad Abu Nofal & Marwan Ashkar
Nova Semiconductors, Inc.

- Datapath generation
- Simple CPU
- Datapath Examples
 - Register file
 - Datapath + Control logic
 - Pipeline registers – clocks
 - Pipeline registers – scan
- Conclusions
- Q&A

Data Path Generation with an ASTRO flow

- When to use data path generation:
 - Where ever custom design of circuits is warranted
 - Any time the design will be ported to another foundry or library
 - On multi dimensional structures:
 - ✓Multi port register files
 - ✓Multipliers
 - ✓Data cross bars
 - Alternatives:
 - DW, PhysOpt, full custom

Benefits of Datapath Generation

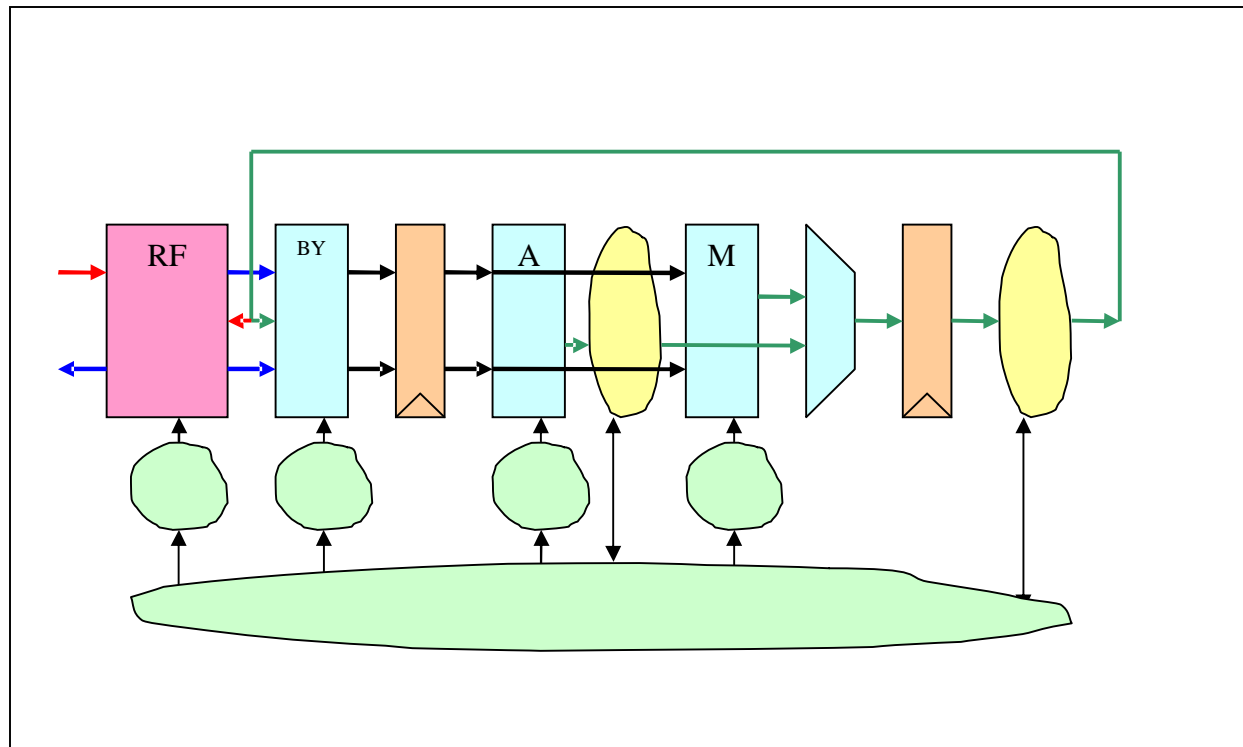
- Similar benefits achieved by Custom design:
 - Very high layout density
 - Lower power – shorter wires
 - Higher speed
 - Can be integrated into ASIC flow as hard macro
- Advantages over Custom design:
 - Lower engineering costs
 - Portable to other foundries and stdcell libraries.
 - Can be integrated into an ASIC flow

Case Study – Media Processor

- Designed a Media processor that integrates:
 - Embedded custom CPU –register file, Caches, Tag RAMs, TLB
 - Powerful DSP, register file, multipliers, adders, cross bars
 - Two media acceleration units – arithmetic calculations
- Integrates several interfaces:
 - DDR, PEX, SATA, USB etc

- Datapath generation
- Simple CPU
- Datapath Examples
 - Register file
 - Datapath + Control logic
 - Pipeline registers – clocks
 - Pipeline registers – scan
- Conclusions
- Q&A

- Block diagram



Simple CPU

-
- Register file (DP + CTRL)
 - Bypass logic (DP + CTRL)
 - Pipeline registers (DP)
 - Arithmetic units (DP + CTRL)
 - Status logic (CTRL)
 - Opcode decoding (CTRL)

- All data path logic was generated using Chipmason (dpGen) from Nova
- All control logic was written in Verilog HDL
- Hierarchical blocks may have DP and CTRL blocks
- DP sub-blocks were read into dc_shell with “dont_touch”
- Placement file was generated (DEF format) for all DP sub-blocks. Empty space was kept for CTRL sections

- Few lines of generated verilog files.....

```
crm_MX4_bS0S1x64Bx4L u1_mx4 (  
    .ab ( in0 ), .bb ( in1 ), .cb ( in2 ), .db ( in3 ), .s0 ( sel1[0] ),  
    .s1 ( sel1[1] ), .yb ( mx4_net1 )  
); // u1_mx4
```

```
crm_SDFFHQ_bx64IncB u0_reg (  
    .clk ( clk ), .ctl ( ck_en0 ), .db ( mx4_net0 ), .qb ( reg_net0 ),  
    .scan_test ( scan_test ), .scanenable ( scanenable ), .si ( si )  
); // u0_reg
```

```
crm_BUF_cx64Bx4L u0_buf (  
    .ab ( reg_net0 ), .yb ( buf_net0 )  
); // u0_buf
```

Verilog => ASTRO

- Importing Verilog into ASTRO

auVerilogIn

```
setFormField "Verilog In Data File" "Verilog File Name" netlist
setFormField "Verilog In Data File" "Verilog List File Name" stublist
setFormField "Verilog In Data File" "Library Name" mainLib
setFormField "Verilog In Data File" "Bus Naming Style" "[%d]"
setFormField "Verilog In Data File"
    "No Backslash Insertion to avoid Hier Name Collisions" "1"
setFormField "Verilog In Data File"
    "Remove All First Backslash Of Escaped Identifier" "1"
formOK "Verilog In Data File"
```

DEF => ASTRO

- Importing DEF file into ASTRO

```
read_def
```

```
  setFormField "read def" "Library Name" mainLib
```

```
  setFormField "read def" "Cell Name" prepCell
```

```
  setFormField "read def" "DEF File" fp_def
```

```
  setFormField "read def" "Physical Input Mode" "1"
```

```
  formOK "read def"
```

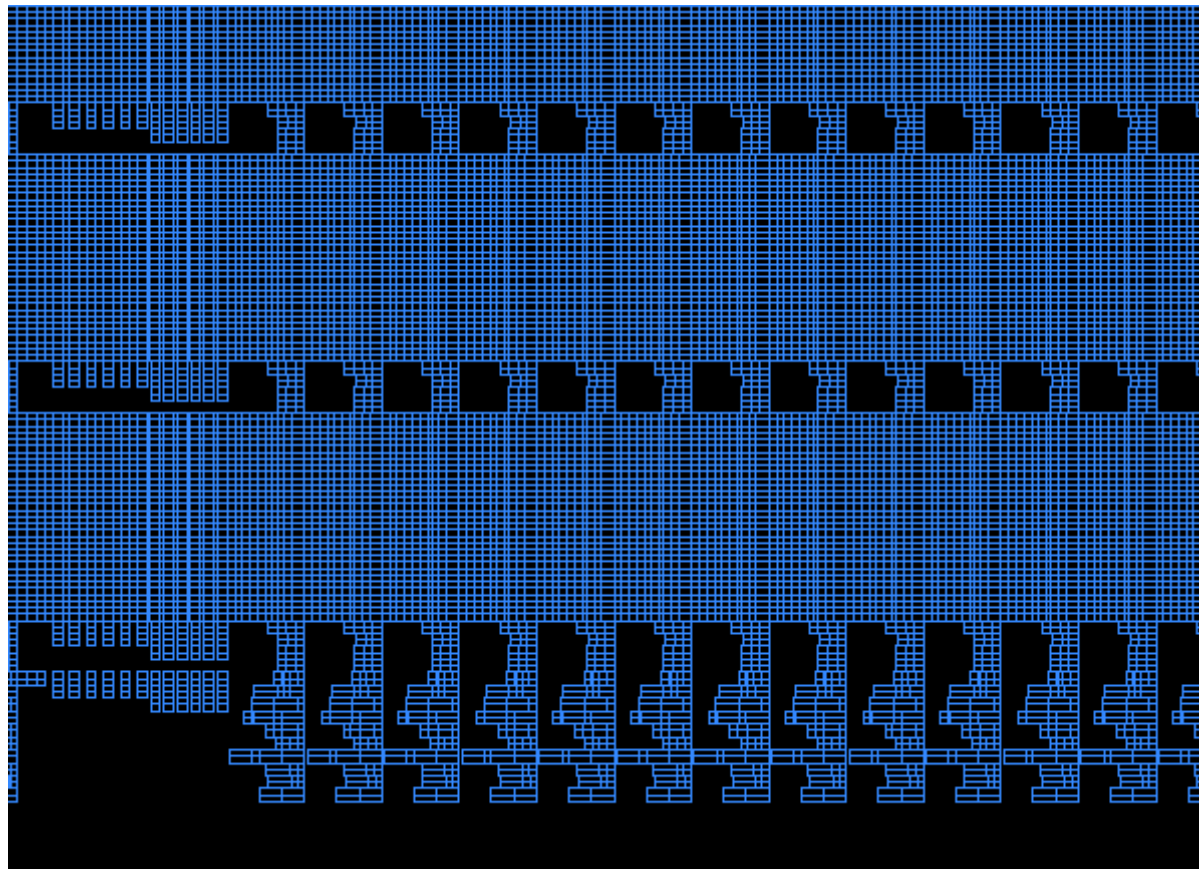
- Datapath generation
- Simple CPU
- **Datapath Examples**
 - Register file
 - Datapath + Control logic
 - Pipeline registers – clocks
- Conclusions
- Q&A

Multi port Register file

- 12 read, 6 write ports, multi threaded very wide register file
- Interleaved 16bits lanes, with 8 rows in between to place drivers
- Read/write select lines are pipelined, placed at the bottom of the array including pre-decoder
- Several other special features: bypass, DfT, and BIST

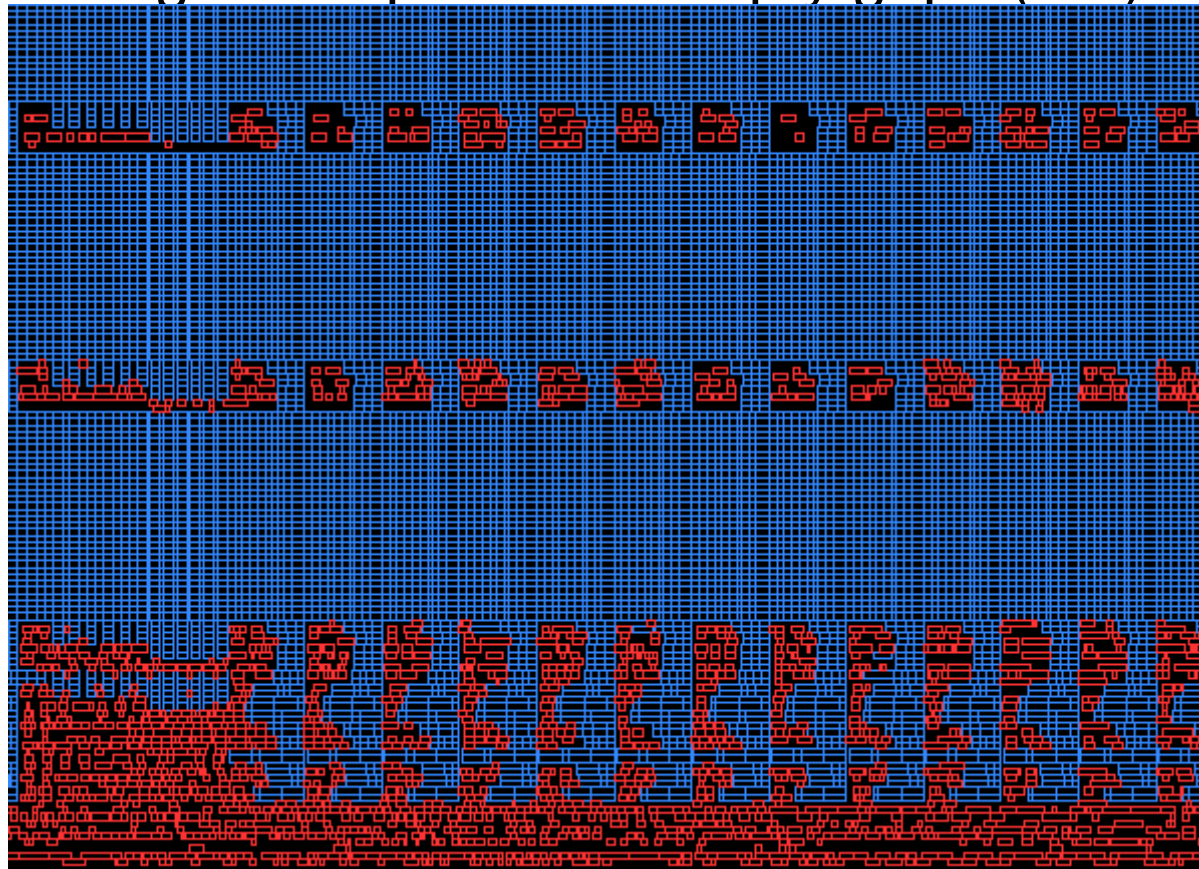
Multi port Register file

- Placement of cells before P&R



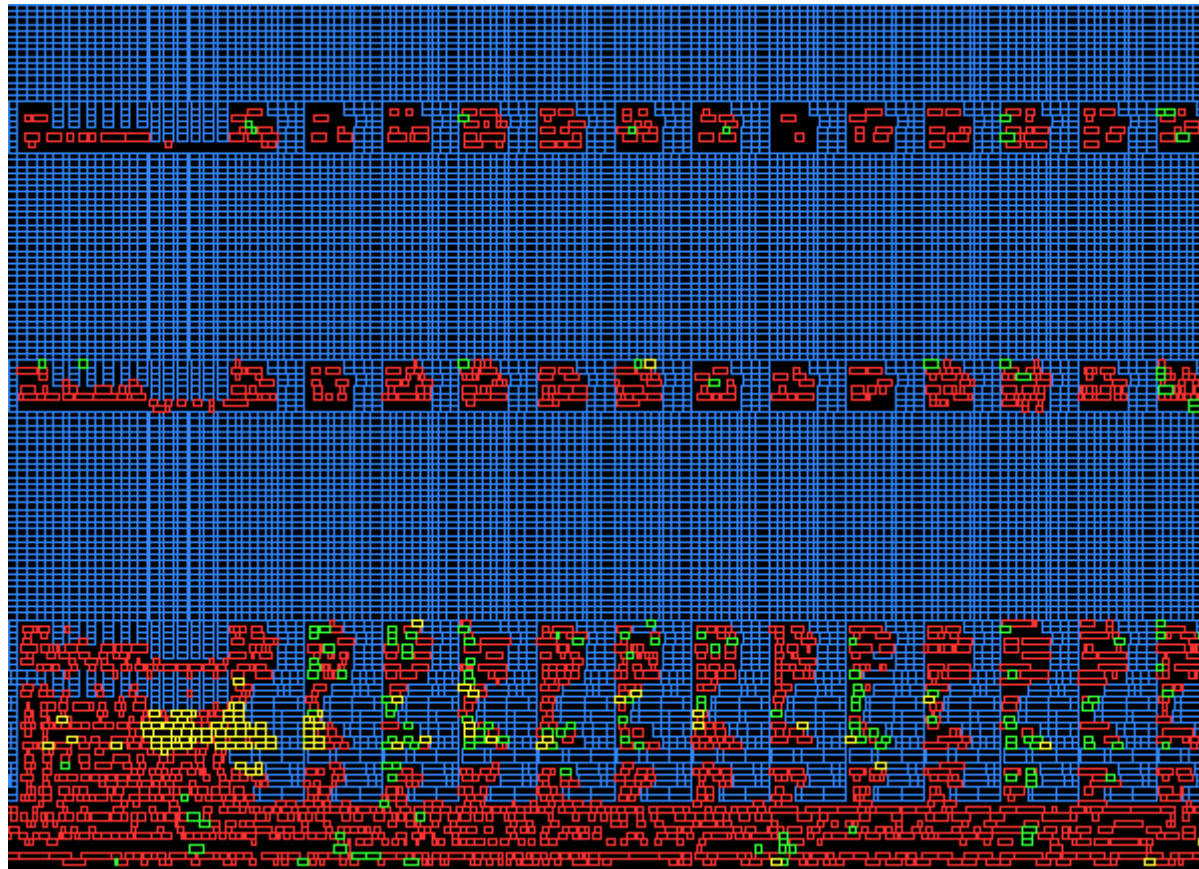
Multi port Register file

- Control logic was placed in empty gaps (red)



Multi port Register file

- After IPO/PPO (green) and CTS (yellow)

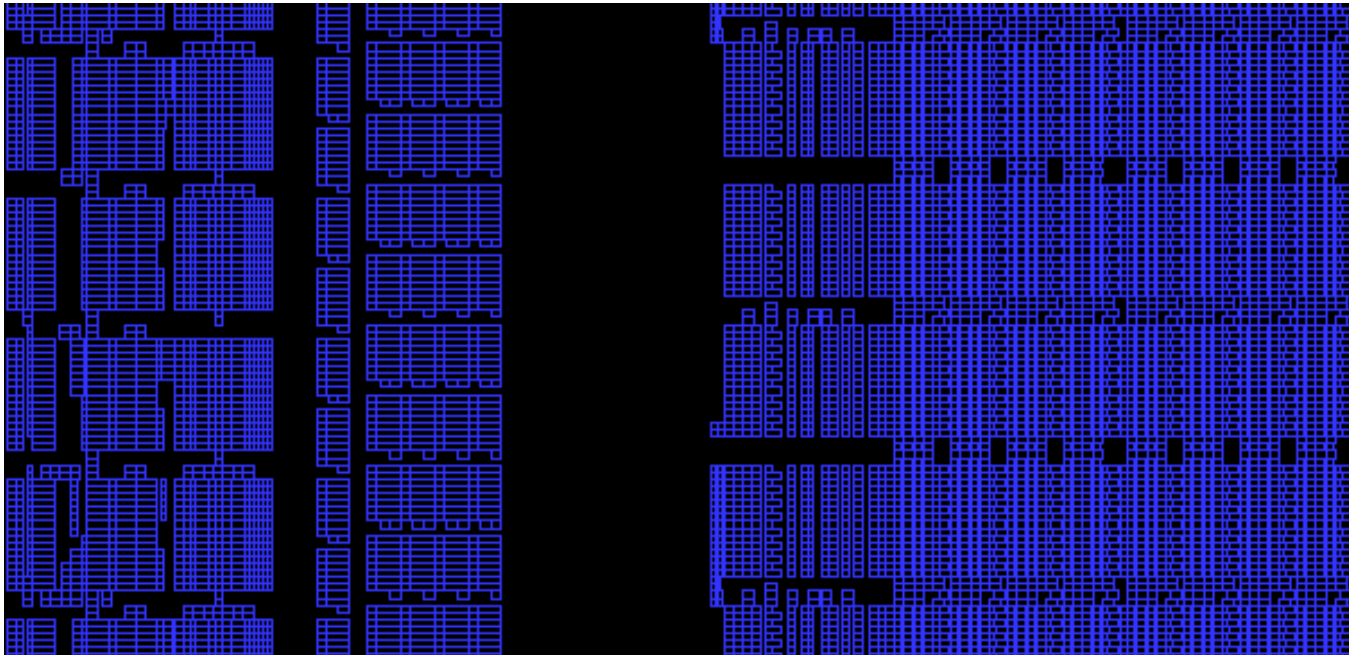


- Datapath generation
- Simple CPU
- **Datapath Examples**
 - Register file
 - **Datapath + Control logic**
 - Pipeline registers – clocks
- Conclusions
- Q&A

- Several 8 and 16 bit data path elements (dpGen)
- Structured multiplexing (dpGen)
 - Followed by “random” logic (HDL/Synthesis)
 - Zero detect
 - Leading one detect
 - Rounding logic
 - Etc.
 - Followed by more Structured logic
- Pipeline registers (dpGen)
- Crossbar (dpGen)

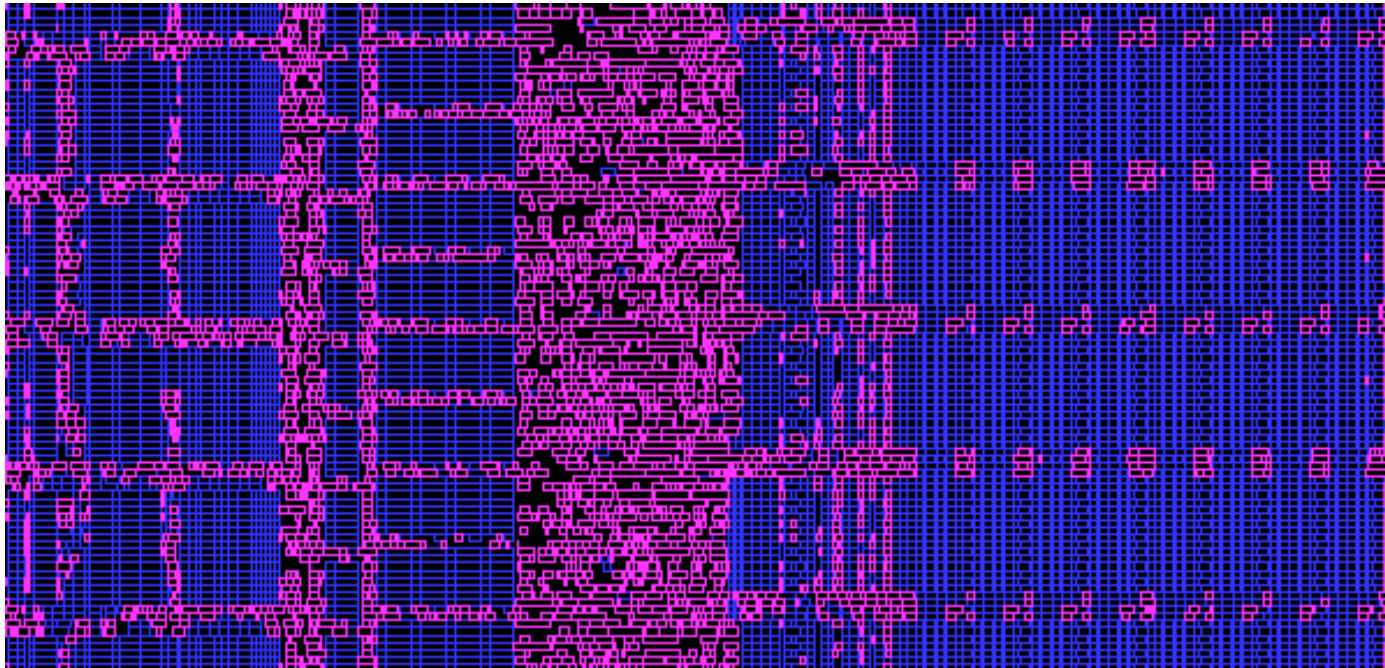
Multi port Register file

- Placement of cells before P&R



Multi port Register file

- Control logic was placed in empty gaps (red)

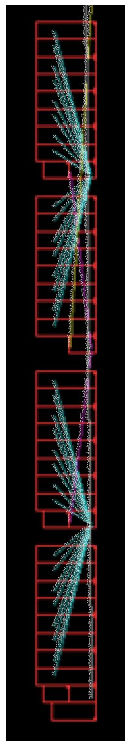


- Datapath generation
- Simple CPU
- Datapath Examples
 - Register file
 - Datapath + Control logic
 - Pipeline registers – clocks
- Conclusions
- Q&A

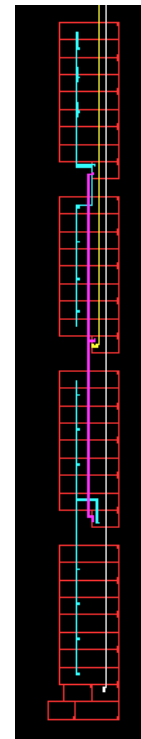
- Interleaved 128-160bit wide
- All bits of the register are controlled by one CTG (Clock Tree Gate)
- A small balanced clock tree is generated and placed after the CTG
 - Clock skew over 256bit wide pipeline register is less than 20ps
- Scan Chain for pipeline register was stitched before placement and was excluded from ASTRO scan stitching

Multi port Register file

- Placement of pipeline register including CTG and clock tree is imported into ASTRO with “FIXED” attribute
- CTS for the whole block is run as usual



Placement,
Fly lines



Placement,
After CTS &
routing

- Datapath generation
- Simple CPU
- Datapath Examples
 - Register file
 - Datapath + Control logic
 - Pipeline registers – clocks
 - Pipeline registers – scan
- **Conclusions**
- Q&A

- Integrated data path generation flow into ASTRO ASIC flow
- Area utilization for data path sections is over 95% for a ~500K instance design of which the datapath is 75% of the design. (50% area saving)
- Improved timing closure (cycle time is 30% better than non datapath approach)
- Repeatable and predictable results
- Significant power reduction as a result of using CTG (Clock Tree Gate) for every pipeline register
 - Saved Power on clock tree and on down stream logic

- Datapath generation
- Simple CPU
- Datapath Examples
 - Register file
 - Datapath + Control logic
 - Pipeline registers – clocks
 - Pipeline registers – scan
- Conclusions
- Q&A