CCSDS Formatting and High Speed SSR Interface

John Dickinson

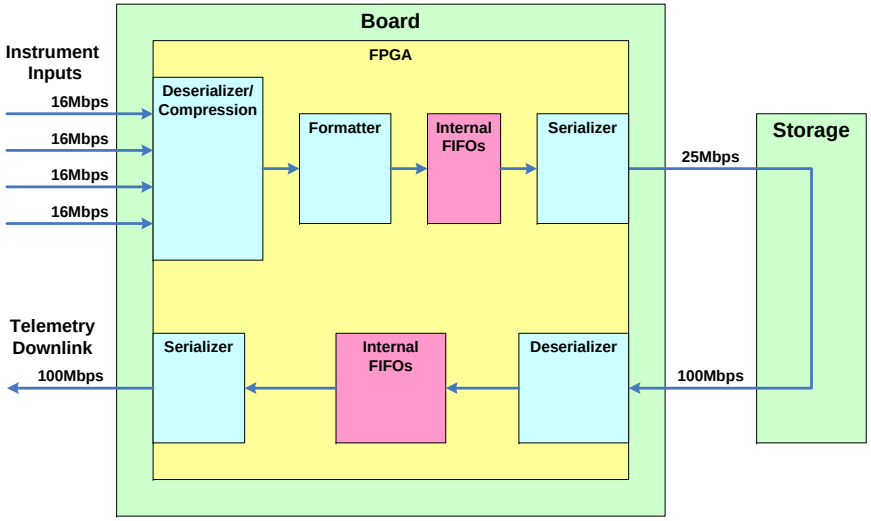
Southwest Research Institute

Abstract

The WISE mission implements high speed serial SSR interfaces coupled with CCSDS formatting. The base design requires the use of multiple internal FIFOs for rate buffering. A series of trades has been performed to determine the most efficient use of the internal RAMs of the Actel RTAX device for future applications requiring large amounts of bandwidth. The maximum performance of the baseline design in an RTAX device will be quantified and the critical paths and bottlenecks identified. Direct comparisons will be made for a variety of internal memory structures, contrasted against external memory devices. Modifications to the design will be limited to utilization of internal RAM (in the form of SMARTGEN macros and user defined modules) and external devices. Maximum performance will be illustrated and the strengths and limitations of various approaches will be highlighted.

Need for Fast Formatting

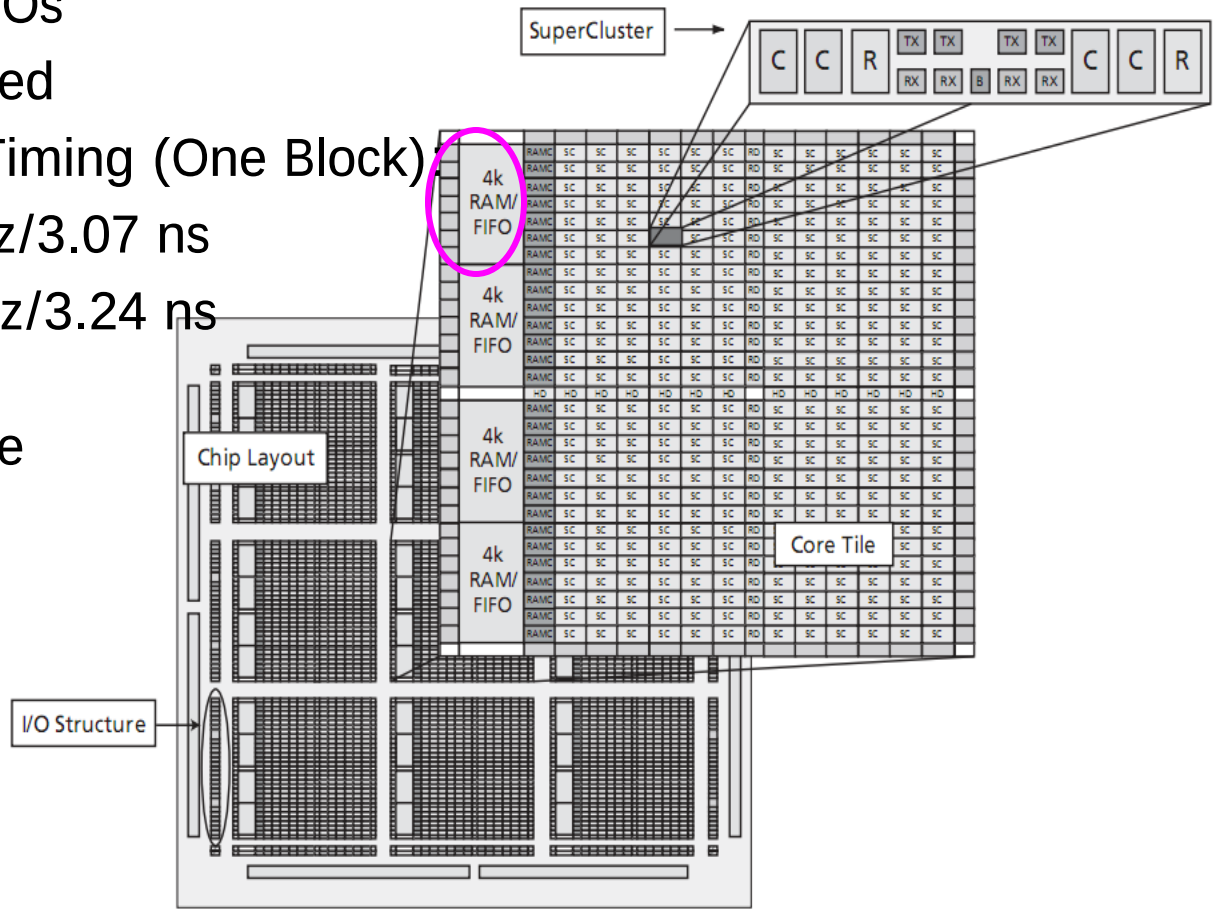
- Instrument interface receives data from instruments at 16 Mbps
- Data is deserialized, compressed, and formatted
- Transmitted to Non-volatile memory at 25 Mbps
- Received from Non-volatile memory at 100 Mbps
- Transmitted to Transponder at 100 Mpbs
- Total Bandwidth:
  - Instrument to Storage:
    - In: 64 Mbps
    - Out: 25 Mbps
  - Telemetry Downlink:
    - In/Out: 100 Mbps



FPGA Characteristics

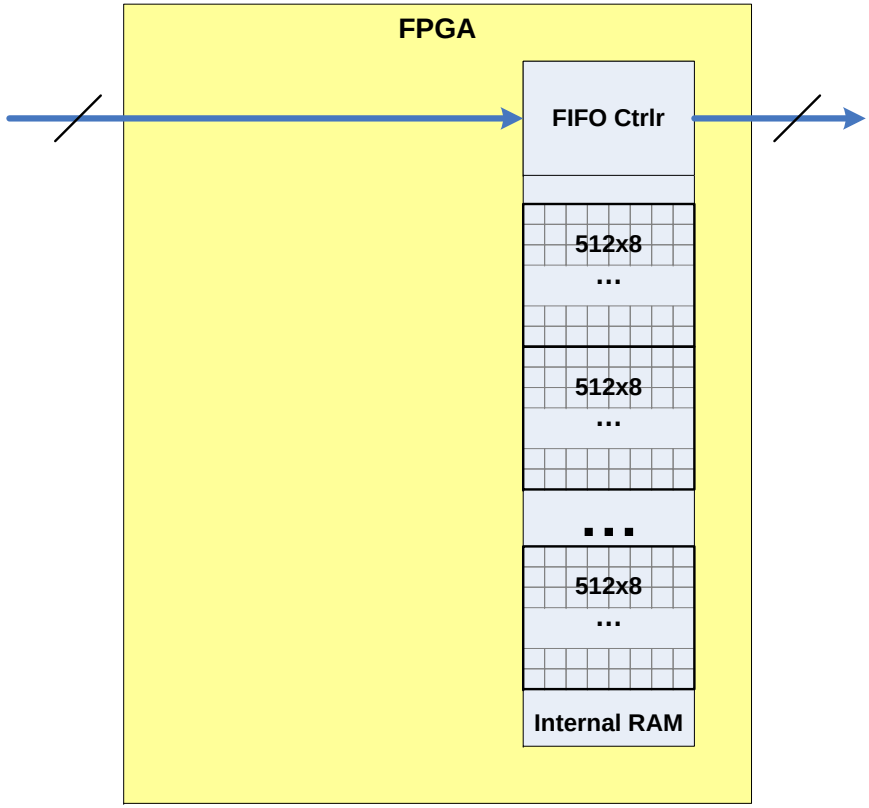
- Actel RTAX2000S-1 FPGA (15% faster than RTAX2000S)
- FIFOs created with pipelined internal SRAMs
- Registers combined with I/Os whenever possible
- Compiled with LVTTL I/Os
- 0 Krad total dose encored
- Advertised RAM Clock Timing (One Block):
  - Read Clk: 325.7 Mhz/3.07 ns
  - Write Clk: 308.6 Mhz/3.24 ns
- In testing the FPGA, it was discovered that the RAMs were causing timing bottleneck

(source: RTAX-S/SL Rad Tolerant FPGAs Datasheet v5.3)



FIFO Architectures 1

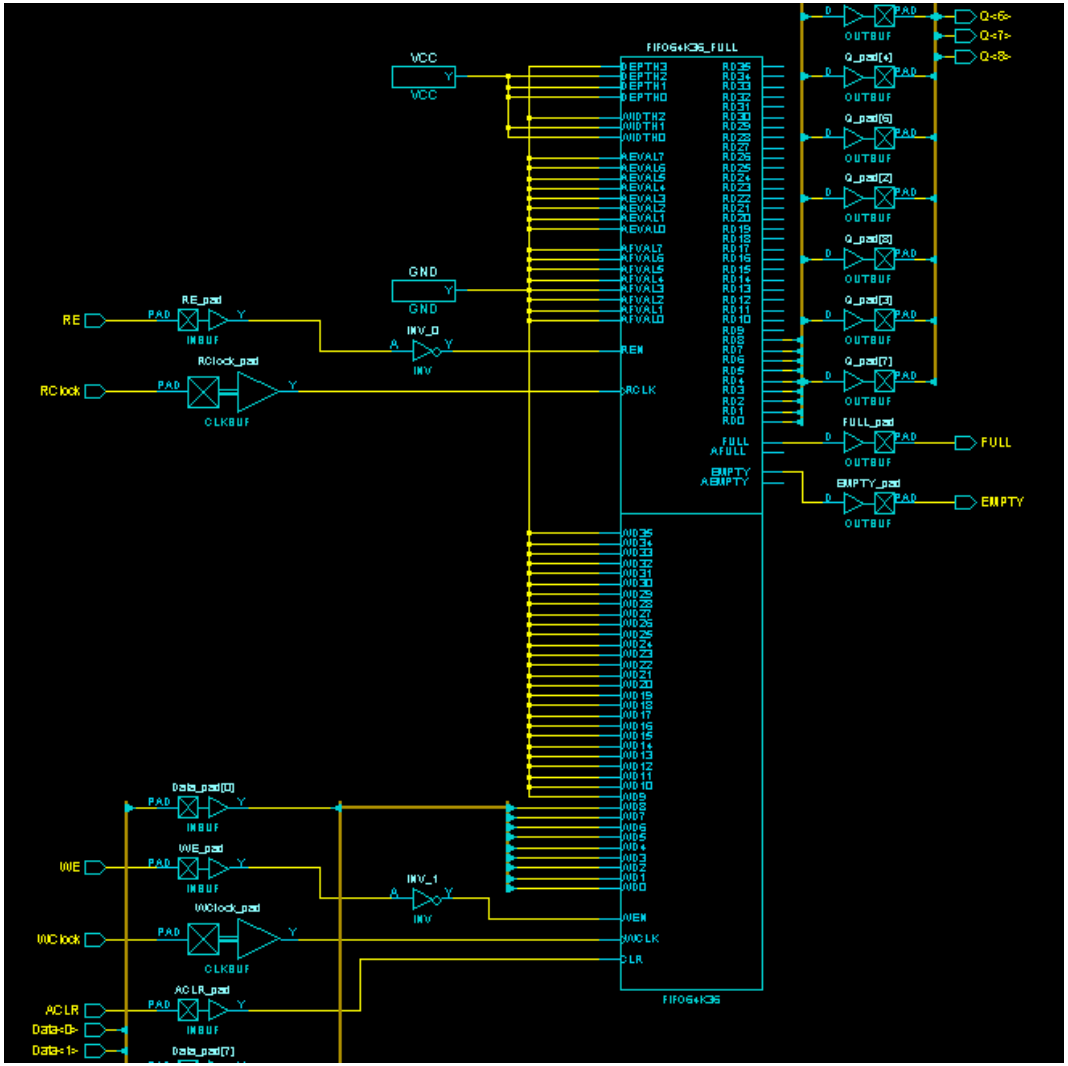
- SmartGen FIFO
  - EMBEDDED Controller
- File Name: FIFO_4096x9.v
- Instantiate: FIFO64K36
- Configure as 8 x (512 x 9) bits
- Synthesis generates FIFO



Synthesis 1

```
Post-Combiner device utilization:
SEQUENTIAL (R-cells)
 Used:    0  Total: 10752   (0.00%)
COMB (C-cells)
  Used:    2  Total: 21504   (0.01%)
LOGIC (R+C cells)
  Used:    2  Total: 32256   (0.01%)
RAM/FIFO
  Used:    8  Total:   64
IO w/Clocks
  Used:   25  Total:  418
CLOCK (Routed)
  Used:    2  Total:    4
HCLOCK (Hardwired)
  Used:    0  Total:    4
PLL
  Used:    0  Total:    0

Input I/O Register  :    0
Output I/O Register :    0
DDR Register        :    0
Comb-Comb (CC)      :    0
Carry Chain         :    0
```



One other note:
Warning: CMP350: This design uses built-in FIFO controllers, which are not SEU enhanced for this device. Actel does not recommend using the built-in FIFO controllers for devices that will be exposed to radiation.

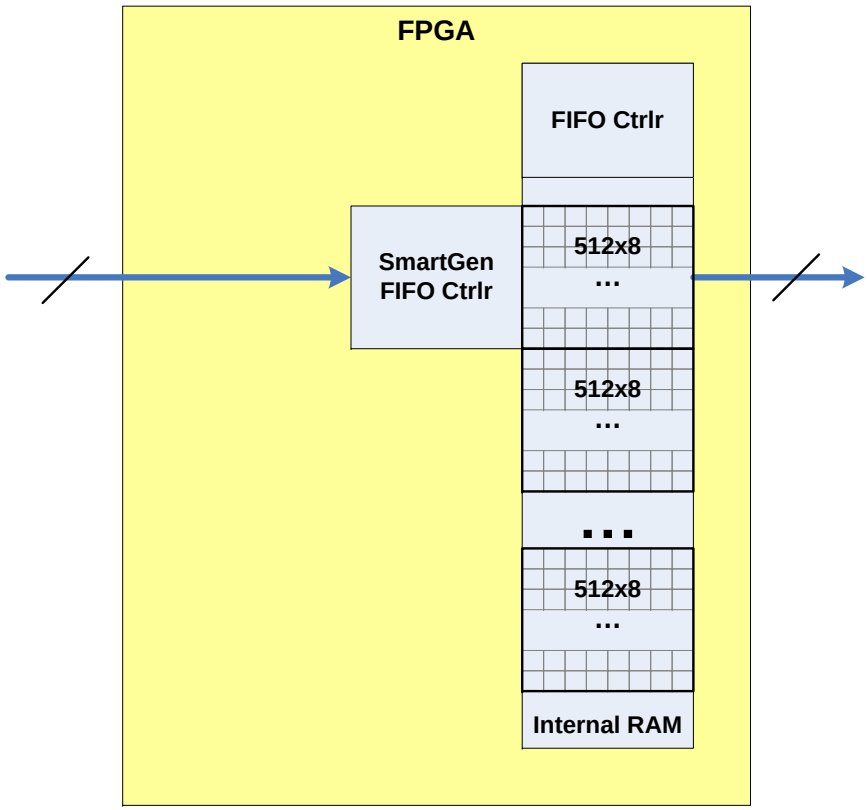
Timing Results 1

- SmartGen FIFO
  - EMBEDDED Controller

Description		Min	Max	Unit
Clock frequency	RClock		89.397	MHz
Clock period	RClock	11.186		ns
Clock frequency	WClock		100.949	MHz
Clock period	WClock	9.906		ns

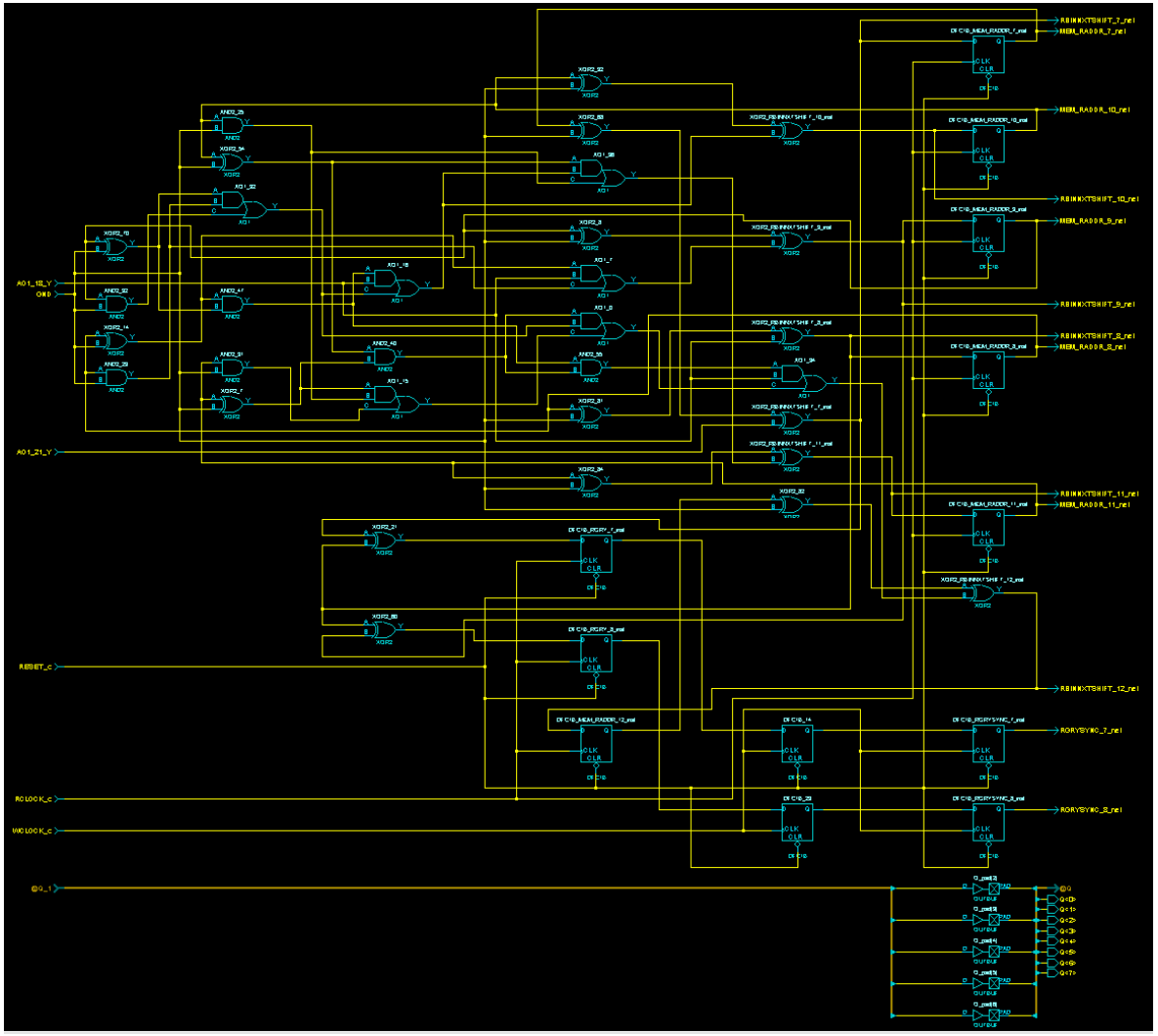
- Results: Not Bad

FIFO Architecture 2



- SmartGen FIFO
  - EXTERNAL Controller
- File Name: FIFOeCtrl_4096x9.v
- Instantiate: RAM64K36
  - and a bunch of logic
- Configure as 8 x (512 x 9) bits
- Synthesis generates Combinatorial Logic FIFO Controller

Synthesis 2



Post-Combiner device utilization:

SEQUENTIAL (R-cells)
Used: 105 Total: 10752 (0.98%)
COMB (C-cells)
Used: 203 Total: 21504 (0.94%)
LOGIC (R+C cells)
Used: 308 Total: 32256 (0.95%)
RAM/FIFO
Used: 8 Total: 64
IO w/Clocks
Used: 25 Total: 418
CLOCK (Routed)
Used: 3 Total: 4
HCLOCK (Hardwired)
Used: 0 Total: 4
PLL
Used: 0 Total: 0

Input I/O Register : 0
Output I/O Register : 11
DDR Register : 0
Comb-Comb (CC) : 0
Carry Chain : 0

Timing Results 2

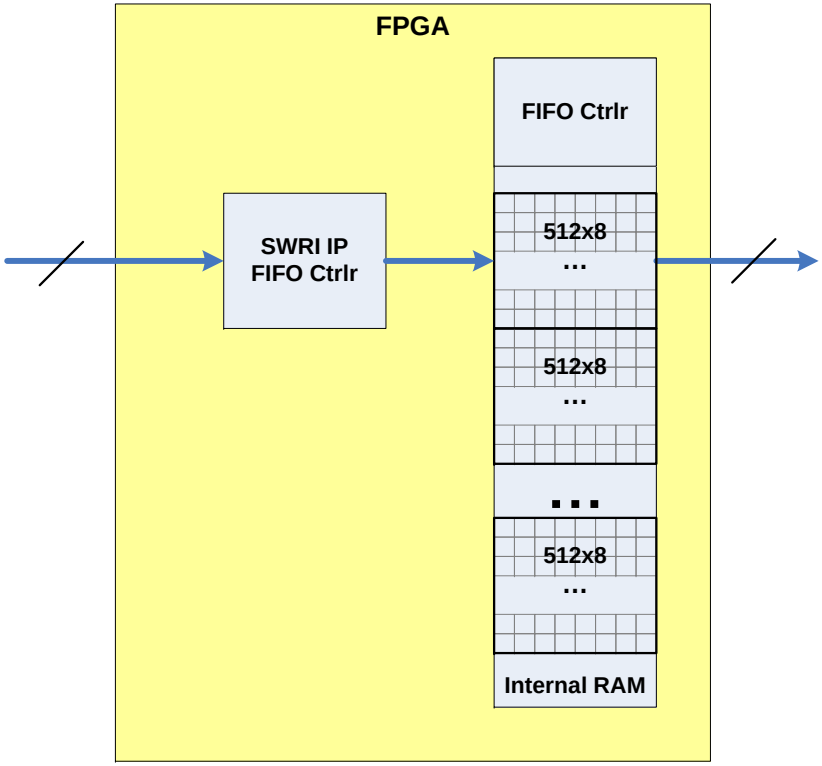
- SmartGen FIFO
  - EXTERNAL Controller

Description		Min	Max	Unit
Clock frequency	RCLOCK		67.504	MHz
Clock period	RCLOCK	14.814		ns
Clock frequency	WCLOCK		62.909	MHz
Clock period	WCLOCK	15.896		ns

- Results: This was NOT an improvement in timing
- However, this design is sufficiently Radiation Hardened to fly
- Synthesized lots of combinatorial logic

FIFO Architecture 3

- SmartGen RAM
  - SWRI IP Controller
- File Name: fifo_8x512x9.v
- Instantiate: RAM64K36P
  - and a bunch of logic
- Configure as 8 x (512 x 9) bits
- Synthesis generated registered FIFO Controller

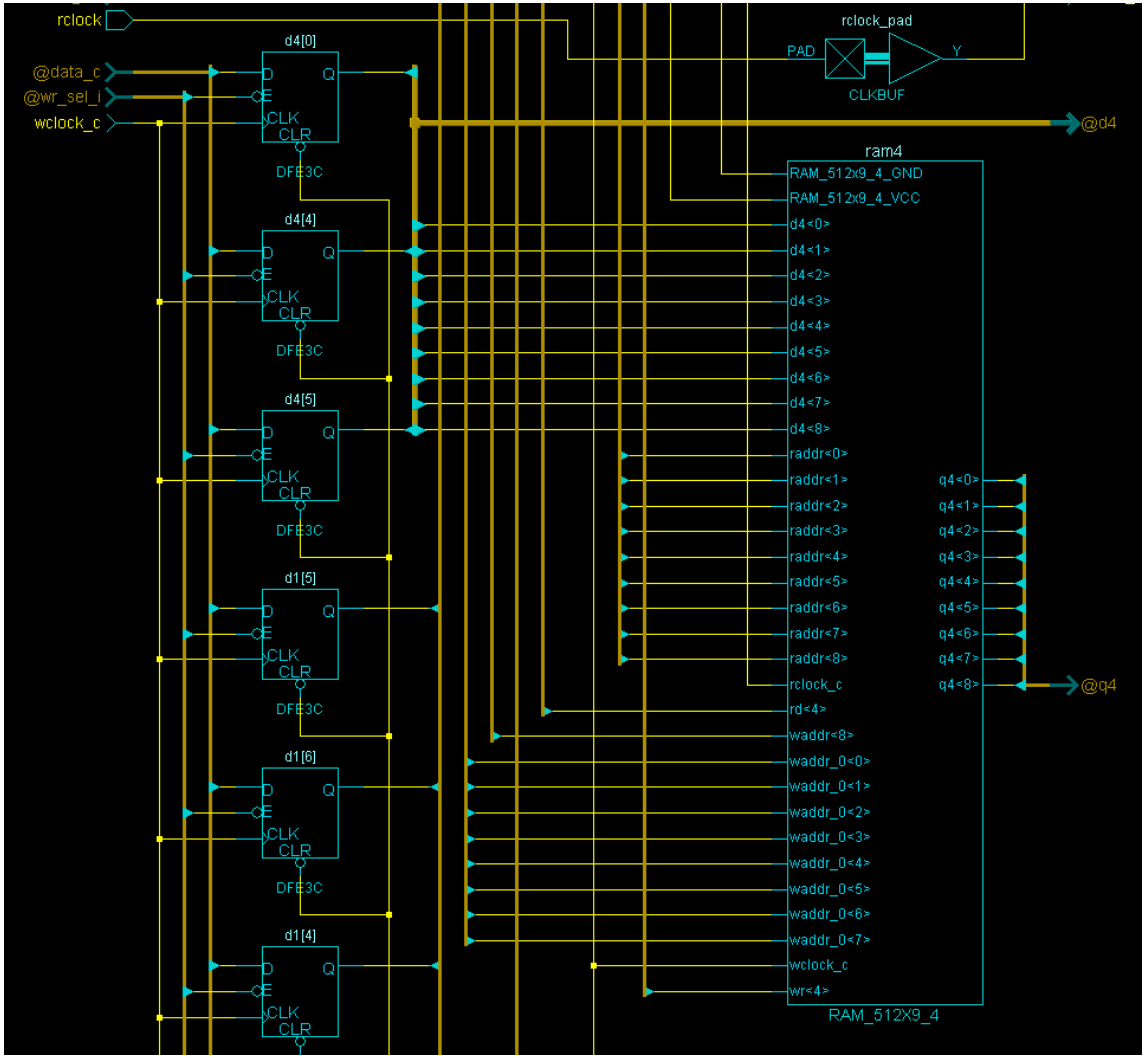


Synthesis 3

Post-Combiner device utilization:

SEQUENTIAL (R-cells)
Used: 157 Total: 10752 (1.46%)
COMB (C-cells)
Used: 111 Total: 21504 (0.52%)
LOGIC (R+C cells)
Used: 266 Total: 32256 (0.82%)
RAM/FIFO
Used: 8 Total: 64
IO w/Clocks
Used: 24 Total: 418
CLOCK (Routed)
Used: 4 Total: 4
HCLOCK (Hardwired)
Used: 0 Total: 4
PLL
Used: 0 Total: 0

Input I/O Register : 0
Output I/O Register : 9
DDR Register : 0
Comb-Comb (CC) : 0
Carry Chain : 0



Timing Results 3

- SmartGen RAM
  - SWRI IP Controller

Description		Min	Max	Unit
Clock frequency	RClock		135.080	MHz
Clock period	RClock	7.403		ns
Clock frequency	WClock		151.286	MHz
Clock period	WClock	6.610		ns

- Results: THIS is an improvement
- Synthesizes more Registers, but less Combinatorial Logic

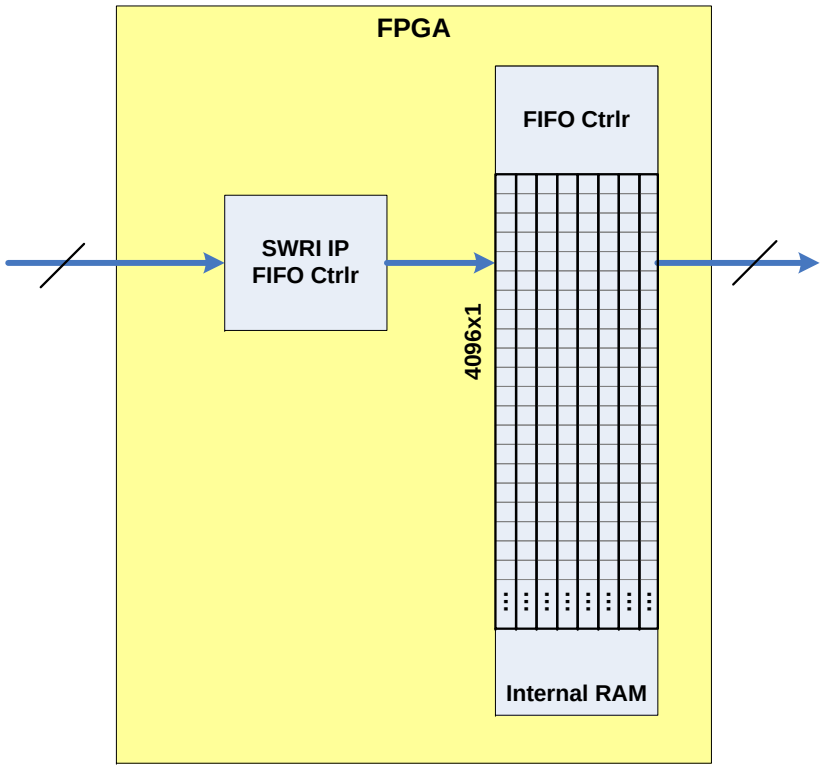
Can we do better?

- Results so far:

CTRL	RAM Dims	Name	WR Freq	RD Freq	R Cells	C Cells
Embedded	NA	FIFO_4096x9.v	100.949	89.397	0	2
SmartGen	NA	FIFOeCtrl_4096x9.v	62.909	67.504	105	203
SWRI IP	512x9	fifo_8x512x9.v	151.286	135.080	157	111

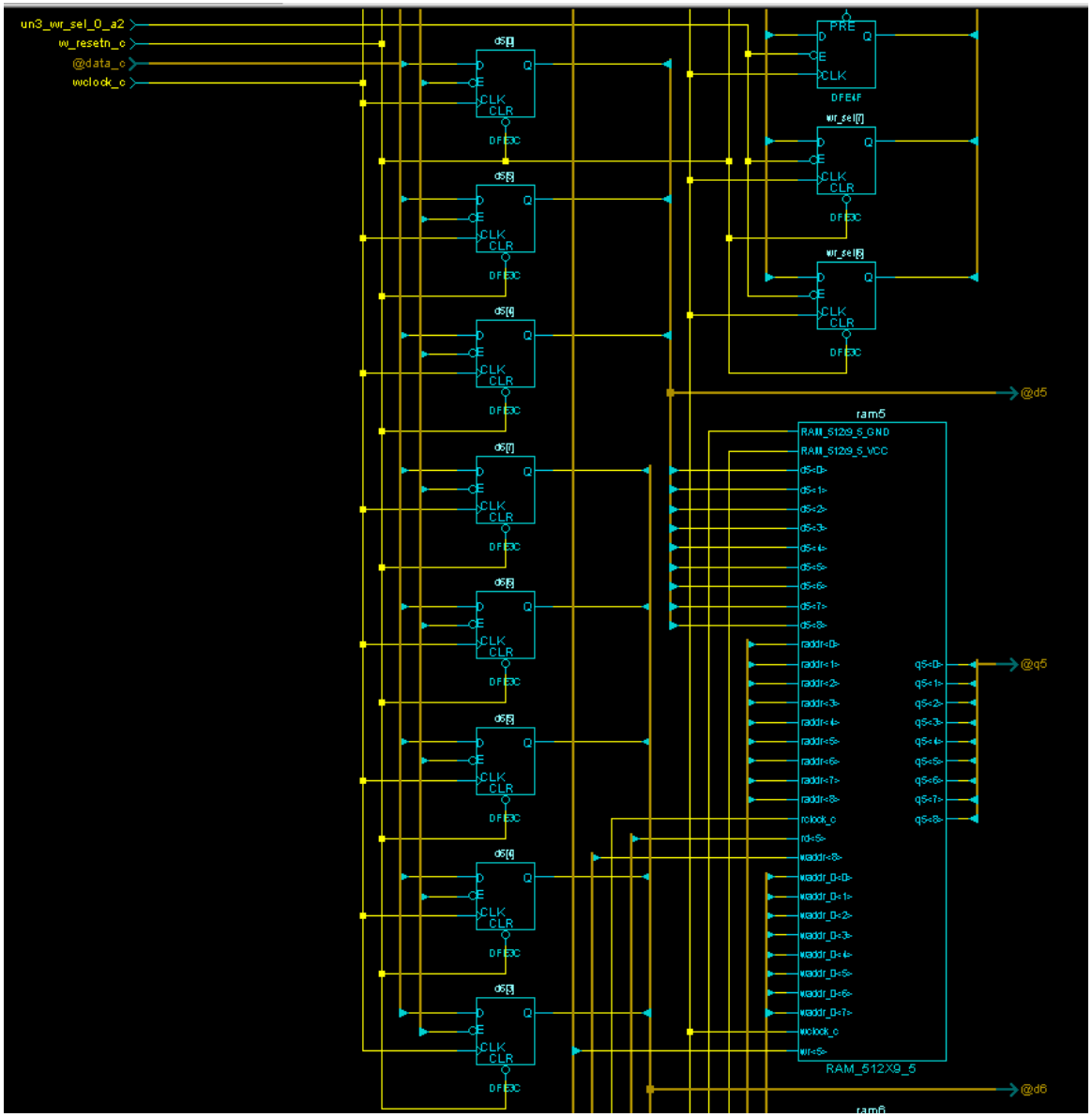
- Winner: SWRI IP FIFO Controller
- We have not changed the depth of the RAMs at all
  - Does this make a difference?

FIFO Architecture 4



- SmartGen RAM
  - SWRI IP Controller
- File Name: fifo_8x4096x1.v
- Instantiate: RAM64K36P
  - and a bunch of logic
- Configure as 8 x (4096 x 1) bits
- Synthesis generated registered FIFO Controller

Synthesis 4



Post-Combiner device utilization:

SEQUENTIAL (R-cells)
Used: 82 Total: 10752 (0.76%)
COMB (C-cells)
Used: 71 Total: 21504 (0.33%)
LOGIC (R+C cells)
Used: 153 Total: 32256 (0.47%)
RAM/FIFO
Used: 9 Total: 64
IO w/Clocks
Used: 24 Total: 418
CLOCK (Routed)
Used: 4 Total: 4
HCLOCK (Hardwired)
Used: 0 Total: 4
PLL
Used: 0 Total: 0

Input I/O Register : 9
Output I/O Register : 9
DDR Register : 0
Comb-Comb (CC) : 0
Carry Chain : 0

Timing Results 4

- SmartGen RAM
  - SWRI IP Controller

Description		Min	Max	Unit
Clock frequency	RClock		125.219	MHz
Clock period	RClock	7.986		ns
Clock frequency	WClock		140.213	MHz
Clock period	WClock	7.132		ns

- Results: Considerably more space efficient
- Not as fast as configuring the RAMs as 512x9

Results

- FIFO with Embedded controller is NOT RAD HARD
- SmartGen FIFO is very resource expensive and VERY slow
- The SWRI IP RAM with 512x9 single RAM access works the best
- The SWRI IP RAM with 4096x1 parallel RAM access is almost as fast, but much smaller, using less than half of the resources.

CTRL	RAM Dims	Name	WR Freq	RD Freq	R Cells	C Cells
Embedded	NA	FIFO_4096x9.v	100.949	89.397	0	2
SmartGen	NA	FIFOeCtrl_4096x9.v	62.909	67.504	105	203
SWRI IP	512x9	fifo_8x512x9.v	151.286	135.080	157	111
SWRI IP	4096x1	fifo_8x4096x1.v	140.213	125.219	82	71

Further Study – Individual FIFOs

- Why is 4096x1 access slower than the 512x9?
  - Is it because of control signals routed to 9 RAMs – high fanout?
- New test: FIFO with 1x(4096x1) RAMs

CTRL	RAM Dims	Name	WR Freq	RD Freq	R Cells	C Cells
SWRI IP	1x4096x1	fifo_1x4096x1.v	156.213	175.439	32	33

- What about 64x36 RAM?
- New test: FIFO with 1x(64x36) RAMs

CTRL	RAM Dims	Name	WR Freq	RD Freq	R Cells	C Cells
SWRI IP	1x64x36	fifo_1x64x36.v	119.119	131.579	55	28

- Uses more resources?
- It is NOT faster
  - Considerably slower than 4096x1 RAM
- So, what is optimal, if 8x512 < 4096x1 < 64x36?

Further Study – Individual FIFOs Results

CTRL	RAM Dims	Name	WR Freq	RD Freq	R Cells	C Cells
SWRI IP	1x4096x1	fifo_1x4096x1.v	156.213	175.439	32	33
SWRI IP	1x2048x2	fifo_1x2048x2.v	188.501	211.015	32	34
SWRI IP	1x1024x4	fifo_1x1024x4.v	228.728	220.022	31	31
SWRI IP	**1x512x9**	**fifo_1x512x9.v**	**234.742**	**220.653**	**35**	**31**
SWRI IP	1x256x18	fifo_1x256x18.v	198.728	202.758	41	28
SWRI IP	1x128x36	fifo_1x128x36.v	119.119	131.579	55	28

- All Aspect Ratios use same FIFO controller
- There is a sweet spot around 512x9

Questions

Further Study – Wide Bus

- Does Timing improve with lower fanout?
- What about accessing RAMs as a wide bus and then “serializing” to 8-bit transactions
- New test: FIFO with 8x(128x36) RAMs

CTRL	RAM Dims	Name	WR Freq	RD Freq	R Cells	C Cells
SWRI IP	8x128x36	fifo_8x128x36.v	141.945	116.117	436	302

- At this point, controller logic becomes too large to make this faster

Further Study – Split Address Buses

- Is timing improved if multiple address buses to the RAMs are used?

CTRL	RAM Dims	Name	WR Freq	RD Freq	R Cells	C Cells
SWRI IP	4096x1	fifo_8x4096x1.v	140.213	125.219	82	71
SWRI IP	4096x1	fifo_8x4096x1maddr.v	138.715	153.775	262	294

- Timing is improved slightly but costs a good chunk of resources