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 Mbps
- 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)
SmartGen FIFO
- EMBEDDED Controller

File Name: FIFO_4096x9.v

Instantiate: FIFO64K36

Configure as 8 x (512 x 9) bits

Synthesis generates FIFO
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.
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
- 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
SmartGen FIFO

- EXTERNAL Controller

<table>
<thead>
<tr>
<th>Description</th>
<th></th>
<th>Min</th>
<th>Max</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clock frequency</td>
<td>RCLOCK</td>
<td></td>
<td>67.504</td>
<td>MHz</td>
</tr>
<tr>
<td>Clock period</td>
<td>RCLOCK</td>
<td>14.814</td>
<td></td>
<td>ns</td>
</tr>
<tr>
<td>Clock frequency</td>
<td>WCLOCK</td>
<td></td>
<td>62.909</td>
<td>MHz</td>
</tr>
<tr>
<td>Clock period</td>
<td>WCLOCK</td>
<td>15.896</td>
<td></td>
<td>ns</td>
</tr>
</tbody>
</table>

Results: This was NOT an improvement in timing

However, this design is sufficiently Radiation Hardened to fly

Synthesized lots of combinatorial logic
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
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
- SmartGen RAM
  - SWRI IP Controller

<table>
<thead>
<tr>
<th>Description</th>
<th></th>
<th>Min</th>
<th>Max</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clock frequency</td>
<td>RClock</td>
<td></td>
<td>135.080</td>
<td>MHz</td>
</tr>
<tr>
<td>Clock period</td>
<td>RClock</td>
<td>7.403</td>
<td></td>
<td>ns</td>
</tr>
<tr>
<td>Clock frequency</td>
<td>WClock</td>
<td></td>
<td>151.286</td>
<td>MHz</td>
</tr>
<tr>
<td>Clock period</td>
<td>WClock</td>
<td>6.610</td>
<td></td>
<td>ns</td>
</tr>
</tbody>
</table>

Results: THIS is an improvement
Synthesizes more Registers, but less Combinatorial Logic
Can we do better?

Results so far:

<table>
<thead>
<tr>
<th>CTRL</th>
<th>RAM Dims</th>
<th>Name</th>
<th>WR Freq</th>
<th>RD Freq</th>
<th>R Cells</th>
<th>C Cells</th>
</tr>
</thead>
<tbody>
<tr>
<td>Embedded</td>
<td>NA</td>
<td>FIFO_4096x9.v</td>
<td>100.949</td>
<td>89.397</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>SmartGen</td>
<td>NA</td>
<td>FIFOeCtrl_4096x9.v</td>
<td>62.909</td>
<td>67.504</td>
<td>105</td>
<td>203</td>
</tr>
<tr>
<td>SWRI IP</td>
<td>512x9</td>
<td>fifo_8x512x9.v</td>
<td>151.286</td>
<td>135.080</td>
<td>157</td>
<td>111</td>
</tr>
</tbody>
</table>

Winner: SWRI IP FIFO Controller

We have not changed the depth of the RAMs at all
  - Does this make a difference?
- 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
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

HCLK (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
■ 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
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.

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

<table>
<thead>
<tr>
<th>CTRL</th>
<th>RAM Dims</th>
<th>Name</th>
<th>WR Freq</th>
<th>RD Freq</th>
<th>R Cells</th>
<th>C Cells</th>
</tr>
</thead>
<tbody>
<tr>
<td>SWRI IP</td>
<td>1x4096x1</td>
<td>fifo_1x4096x1.v</td>
<td>156.213</td>
<td>175.439</td>
<td>32</td>
<td>33</td>
</tr>
</tbody>
</table>

- What about 64x36 RAM?

- New test: FIFO with 1x(64x36) RAMs

<table>
<thead>
<tr>
<th>CTRL</th>
<th>RAM Dims</th>
<th>Name</th>
<th>WR Freq</th>
<th>RD Freq</th>
<th>R Cells</th>
<th>C Cells</th>
</tr>
</thead>
<tbody>
<tr>
<td>SWRI IP</td>
<td>1x64x36</td>
<td>fifo_1x64x36.v</td>
<td>119.119</td>
<td>131.579</td>
<td>55</td>
<td>28</td>
</tr>
</tbody>
</table>

- 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

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

- 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

<table>
<thead>
<tr>
<th>CTRL</th>
<th>RAM Dims</th>
<th>Name</th>
<th>WR Freq</th>
<th>RD Freq</th>
<th>R Cells</th>
<th>C Cells</th>
</tr>
</thead>
<tbody>
<tr>
<td>SWRI IP</td>
<td>8x128x36</td>
<td>fifo_8x128x36.v</td>
<td>141.945</td>
<td>116.117</td>
<td>436</td>
<td>302</td>
</tr>
</tbody>
</table>

- At this point, controller logic becomes too large to make this faster
Is timing improved if multiple address buses to the RAMs are used?

<table>
<thead>
<tr>
<th>CTRL</th>
<th>RAM Dims</th>
<th>Name</th>
<th>WR Freq</th>
<th>RD Freq</th>
<th>R Cells</th>
<th>C Cells</th>
</tr>
</thead>
<tbody>
<tr>
<td>SWRI IP</td>
<td>4096x1</td>
<td>fifo_8x4096x1.v</td>
<td>140.213</td>
<td>125.219</td>
<td>82</td>
<td>71</td>
</tr>
<tr>
<td>SWRI IP</td>
<td>4096x1</td>
<td>fifo_8x4096x1maddr.v</td>
<td>138.715</td>
<td>153.775</td>
<td>262</td>
<td>294</td>
</tr>
</tbody>
</table>

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