QUEENS-FPGA: Quality Evaluation of European New SW for BRAVE FPGA

SEFUW 2018
ESTEC 11th April 2018

David Gonzalez-Arjona – FPGA Design Engineer
Tres Cantos – Madrid (Spain)
Contact: dgarjona@gmv.com
NanoXmap SW Tools

QUEENS-FPGA at SEFUW18
QUEENS-FPGA OBJECTIVES

**QUality assessment Evaluation of European NanoXplore SW for brave FPGA**

- BRAVE project provided very promising European SRAM-based FPGAs for Space
- FPGA Development SW is really important to take full advantage of the technology
  - Quality systematic Assessment Plan based on measurable FPGA SW Tools Metrics identification
  - HW benchmarking identification based on consortium background knowledge (GMV, NTUA, NanoXplore)
  - Performance and Reliability Evaluation
  - FPGA and HDL-languages expertise on Space projects under ESA contracts
  - Migration or Porting to BRAVE of in-house developed IP-core for space project
EVALUATION RESULTS BASES

DOES IT MAKE THE JOB?

Can you use it for your mission?

SW tools comparison

Usability, Flexibility, Reporting, Inference, Libraries

HW devices comparison

Power, Performances, Area

QUEENS-FPGA at SEFUW18
QUEENS-FPGA METRICS

ABILITY TO:
- synthesize a RTL code
- ability to route a design
- Inferred VHDL coding and mapping

QUALITY OF REPORTS
- Organization
- Synthesis estimations
- Timing reports

DESIGN REUSE
- IP cores provided
- generating new IP cores
- Support of reuse standards

ADVANCED DESIGN FLOWS
- Floorplanning
- Incremental design
- Third-Party Integration (Simulation)

SW TOOL PC CONSTRAINTS:
- Memory resources
- Execution time
- CPU usage

USER EXPERIENCE
- Ease of use
- Customization of the tools
- Scripting capabilities

HARDWARE INTERACTION
- Programming time
- verification of bitstream
- hardware debugging

MAIN DEV. PRODUCT METRICS
- RTL Optimization
- Area vs Timing vs Power
- Constraints

QUEENS-FPGA at SEFUW18
QUEENS-FPGA METHODOLOGY

• Follow a systematic & unified approach throughout the tool assessment project
  1. metrics & methodology (Task1) ✔
  2. benchmark selection (Task1) ✔
  3. synthesis (Task2) ✔
  4. Place & Route (Task3) ✔
  5. 3rd party tools (Task4)
  6. HW testing (Task5)

• Different complexity level circuits
• Fair selection of comparable devices
  - Technology
  - Resources
  - Hardened
• Methodology serves a twofold purpose
  1. explore/examine multiple tool issues
     • parameters, load, trade-offs, etc.
     • compare to 3rd tools, show quality
  2. avoid dead ends & erroneous reports
     • use feedback paths, simulations
QUEENS-FPGA at SEFUW18

QUEENS-FPGA SYNTHESIS

- Built in same technology node 65 nm, SRAM-based

- Similar in architecture/technology
  - 4 input LUTs
  - DSPs (18*18 bits)
  - Registers
  - DFFs

- Similar available resources
  - Embedded RAM blocks
  - Number of LUTs, DSPs, DFF

- Space-grade devices
  - Rad-hard High-performance devices (not so many)

- Each devices are supported by its vendor SW tool

- NanoXmap 2.6.3 -> NanoXmap 2.8.4
  (Not yet tested 2.8.5)

- Trade-off
**BENCHMARK CIRCUITS**

### SIMPLE CIRCUITS (DC1):
- Light in terms of area utilization
- Light in effort requirement for the SW tools
- The basics
- Simplicity

### MID-COMPLEX CIRCUITS (DC2):
- Heavier in terms of area utilization
- More exigent for the SW tools
- Less complex than DC3

### COMPLEX CIRCUITS (DC3):
- Most complex of the designs
- Worth to elaborate a specific section for the selection
- Stress the FPGA synthesis challenges

### PORTING:
- Image feature detection & tracking
- Re-design to fit in NG-MEDIUM up to the limits of resources

<table>
<thead>
<tr>
<th>Bench-mark</th>
<th>Id</th>
<th>Configuration</th>
<th>Bench-mark</th>
<th>Id</th>
<th>Configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multiplexor</td>
<td>DC1-01-C1</td>
<td>Case sentence</td>
<td>VGA controller</td>
<td>DC1-07-C1</td>
<td>Default</td>
</tr>
<tr>
<td>Multiplexor</td>
<td>DC1-01-C2</td>
<td>If-else sentence</td>
<td>LED controller</td>
<td>DC1-08-C1</td>
<td>Default</td>
</tr>
<tr>
<td>Multiplexor</td>
<td>DC1-01-C3</td>
<td>Concurrent sentence</td>
<td>32-bits FPM</td>
<td>DC1-09-C1</td>
<td>Without DSP</td>
</tr>
<tr>
<td>4 bit counter</td>
<td>DC1-02-C1</td>
<td>Asynchronous reset</td>
<td>32-bits FPM</td>
<td>DC1-09-C2</td>
<td>With DSP</td>
</tr>
<tr>
<td>4 bit counter</td>
<td>DC1-02-C2</td>
<td>Synchronous reset with priority enable</td>
<td>FIFO buffer</td>
<td>DC1-10-C1</td>
<td>18Kx18</td>
</tr>
<tr>
<td>4 bit counter</td>
<td>DC1-02-C3</td>
<td>Synchronous priority reset with enable</td>
<td>CAN controller</td>
<td>DC2-01-C1</td>
<td>Default</td>
</tr>
<tr>
<td>BCD counter</td>
<td>DC1-03-C1</td>
<td>Standard VHDL design method</td>
<td>LEON 3</td>
<td>DC2-02-C1</td>
<td>Basic</td>
</tr>
<tr>
<td>BCD counter</td>
<td>DC1-03-C2</td>
<td>2 process VHDL design method</td>
<td>Guess game</td>
<td>DC2-03-C1</td>
<td>Default</td>
</tr>
<tr>
<td>Register bank</td>
<td>DC1-04-C1</td>
<td>Case sentence</td>
<td>SDRAM controller</td>
<td>DC2-04-C1</td>
<td>Default</td>
</tr>
<tr>
<td>Register bank</td>
<td>DC1-04-C2</td>
<td>Indexed array</td>
<td>SpaceWire</td>
<td>DC2-05-C1</td>
<td>Default</td>
</tr>
<tr>
<td>Finite state machine</td>
<td>DC1-05-C1</td>
<td>If-else sentence</td>
<td>FIR</td>
<td>DC3-01-C1</td>
<td>Configuration 2</td>
</tr>
<tr>
<td>Finite state machine</td>
<td>DC1-05-C2</td>
<td>One-Hot encoding</td>
<td>Harrys</td>
<td>DC3-02-C1</td>
<td>Configuration 3</td>
</tr>
<tr>
<td>PWM</td>
<td>DC1-06-C1</td>
<td>Without DSP</td>
<td>Disparity</td>
<td>DC3-03-C1</td>
<td>Configuration 1</td>
</tr>
<tr>
<td>PWM</td>
<td>DC1-06-C2</td>
<td>With DSP</td>
<td>VBN</td>
<td>Porting</td>
<td>Ad-hoc</td>
</tr>
</tbody>
</table>
FIRST RESULTS DC1

- Overall increased number of LUTs used in low complexity designs
  - Mainly, due to use of these as pass-thru logic and suboptimal performance of arithmetic operations.

- Less maximum frequency achieved

- Efficient use of DFFs, DSPs and RAMBs in low complexity circuits

- Very low dynamic power consumption reported

- Place and route time over reference value, but still acceptable taking into account that ‘Timing driven’ option has been enabled in all DC1 circuits
**FIRST RESULTS DC2**

- Overall increased number of LUTs used in medium complexity designs
  - Mainly, because of the use of these as pass-thru logic and routed elements.
- Less maximum frequency achieved \([-30\% \text{ to } +50\%]\)
- Efficient use of DFFs
- Very low dynamic power consumption reported

- Better maximum frequencies are not always achieved with 'Timing driven' option than without it
- *Timing driven* option increases place and route time around 45%
1. NG-MEDIUM technology and specific hardware resources

2. NG-MEDIUM technology added to gencomp pkg with following features:
   - FPGA device
   - SRAM resources (Embedded block RAMs and Register file blocks)
   - Unavailable simultaneous read & write operations to same address in BRAMs
   - 2 ports BRAMs
   - True dual port BRAMs
   - Pads
   - Inverted output enable pad polarity
   - Embedded clock and resources (Clock switch component – CKS)
   - Embedded clock generator resources (phase-locked loop component – PLL)

3. Clock generator of LEON 3 with NG-MEDIUM clock resources (PLLs and WFGs-Wave form generators). Clock and functionality with NG-MEDIUM resources (CKS).

4. Different configurations of BRAMs are instantiated related with the size of data and address bus. In the case of memories up to 64 words the design instantiates register file blocks (RFBs) instead of BRAMs.

5. IO pads of LEON3 with the IO resources of NG-MEDIUM.

6. New technology is integrated in the GRLIB. Modify all files using the adapted components. Add IF_GENERATE

7. Different hardware in the board like DDR memories or PHY interfaces may need independent IPs to control the hardware.
PORTING VISION-BASED NAVIGATION IP

- NG-MEDIUM device has limited resources
- Starting point: (1) synthesize the design as it is
- Check the result
  - If it fits, go ahead
  - If it does not fit, (2) make it fit:
    - Modify configurations of the design
    - HW/SW redesign part of the project
- Once it fits, (3) modify the design to optimise it
- Consider different communication protocols and interfaces to transfer data between board and host
- Implement the chosen interface (UART) and check it
- Add and manage communication interface to RelNav IP core
- Implement full design and take metrics
RESULTS OF VISION-BASED NAVIGATION

Options:
- UseNxLibrary: Yes
- DefaultFSMEncoding: Binary
- DefaultRAMMapping: RAM
- DefaultROMMapping: LUT
- MaxRegisterCount: 10100
- ManageUnconnectedOutputs: Error
- ManageUnconnectedSignals: Error
- TimingDriven: No

MappingEffort:
- High
- MergeRegisterToPad: Never
- AdderToDSPMapThreshold: 0
- LessThanToDSPMapThreshold: 0
- MultiplierToDSPMapThreshold: 1

Last Version of NanoXmap v2.8.5

<table>
<thead>
<tr>
<th>Metric</th>
<th>NanoXmap</th>
<th>Vendor 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>LUTs</td>
<td>19402</td>
<td>7386 (6-in)</td>
</tr>
<tr>
<td>DFFs</td>
<td>7397</td>
<td>3526</td>
</tr>
<tr>
<td>DSPs</td>
<td>21</td>
<td>16</td>
</tr>
<tr>
<td>BRAMs</td>
<td>37</td>
<td>58</td>
</tr>
<tr>
<td>Total Memory (Kbits)</td>
<td>1776</td>
<td>1044</td>
</tr>
<tr>
<td>Max. Freq (MHz)</td>
<td>39</td>
<td>145</td>
</tr>
<tr>
<td>Dynamic Power Consumption</td>
<td>0,37W</td>
<td>1,33W</td>
</tr>
<tr>
<td>Static Power Consumption</td>
<td>0,73W</td>
<td>3,5W</td>
</tr>
</tbody>
</table>
- Very preliminary Demo, cheating just a bit:
  - Not in real-time
  - Not in closed loop
  - HW/SW split
PRELIMINARY CONCLUSIONS

• Architecture Matters: Learn how to optimize design for the new board architecture
• Porting → wrappers over specific HW resources (DSPs, BRAMs)
• Very responsive Vendor support team.
• Improvements version by version
  • Corner cases or different improvement targets make some parameter worse
• NG-MEDIUM is in the market → NanoXplore is an actual competitor → Portfolio increasing → Good for Industry

- Great improvement of generated reports in 2.8.3 version
- Efficient use of DFFs in low and medium complexity designs
- Reduced use of LUTs and DFFs when RAMB or DSP are used in low and medium complexity designs.
- Very low dynamic power consumption reported
- Competitive implementation times

- Increased use of LUTs compared to 3rd party tools ~[+10% to +60%]
- Lower maximum frequencies achieved ~[+30% to +50%]
- Less flexibility to tailor elements in designs

* Depends on reporting (LUTs for HDL, LUTs pass-through) [architecture matters].
MORE FROM QUEENS-FPGA

Application to NanoXplore
FPGA SW tools
THANK YOU

David Gonzalez-Arjona