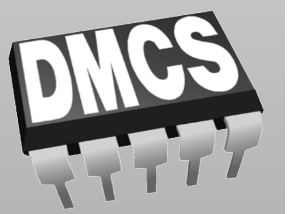


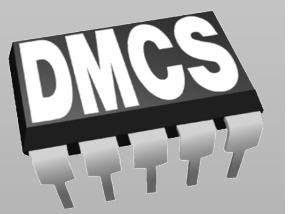
DSP development for uTC

Paweł Prędki
pawel.predki@desy.de

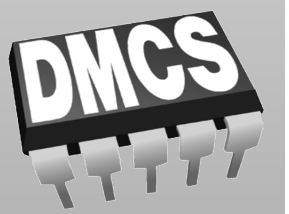
Department of Microelectronics and Computer Science
Technical University of Lodz

LLRF Collaboration Workshop 2011
Warsaw, 14-16 December 2011

Outline

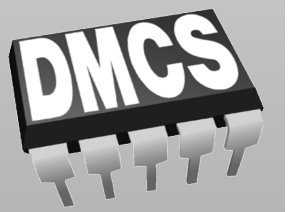


Outline



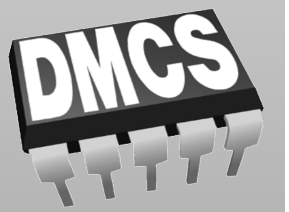
Project history

- Development started on Simcon DSP years ago
- I resumed the project using Simcon DSP (fall 2010), ATCA Carrier Board and ultimately uTC
- Initial plans to use it as an aid to the controller on Simcon
- Evolved into a DSP development framework to test DSP usefulness for LLRF applications

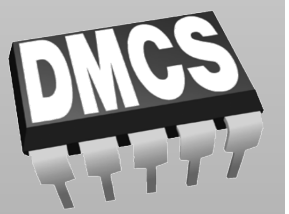


Potential use

- General idea - aid the uTC controller in algorithm calculations
- Better precision thanks to floating-point capabilities
- Potential algorithms to perform in the DSP:
  - Cavity detuning
  - Learning feed-forward
  - IQ calculation



Outline

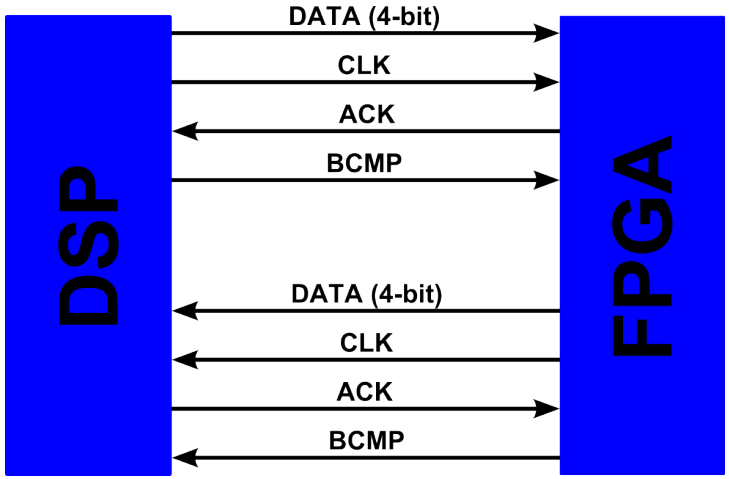


Basic specs of ADSP TS201

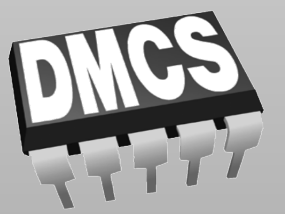
- Floating-point unit
- Clock frequency up to 600 MHz - currently clocked with 100 MHz - with an instruction cycle of 1.67 ns
- 24 Mbits of internal DRAM memory
- Up to four ASM instructions per clock cycle but quite restrictive
- Four full-duplex LVDS link ports

Fast serial link ports

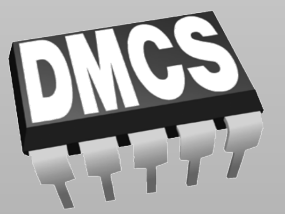
- Differential clock and four differential data lines
- Data transmitted on BOTH clock edges
- Basic unit of transmission - 1 quad = 128 bits
- Maximum theoretical throughput @100 MHz - 800 Mbps; obtained 630 Mbps (raw data loopback)
- Used for DSP booting
- Three links connected to uTC



Outline



VHDL DSP communication module



- Fully integrable with a larger VHDL project, e.g. controller
- Takes care of everything - only requires connection to a data input feed and a data output sink
- Two modes of operation:
  - Raw data mode - no data verification, no overhead
  - Communication protocol mode - uses a protocol defined for DSP $\leftrightarrow$ FPGA communication. Checksum verification, transmission reliability, overhead

Communication protocol

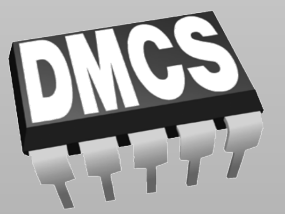
Frame header
Data 0
Data 1
…
Data 15

Fig. 1. Communication protocol frame

127 120	119 112	96 89	88 73	39 32	31 0
Algorithm ID	Algorithm subtype	Words in frame	Words per algorithm	Debug character	Checksum

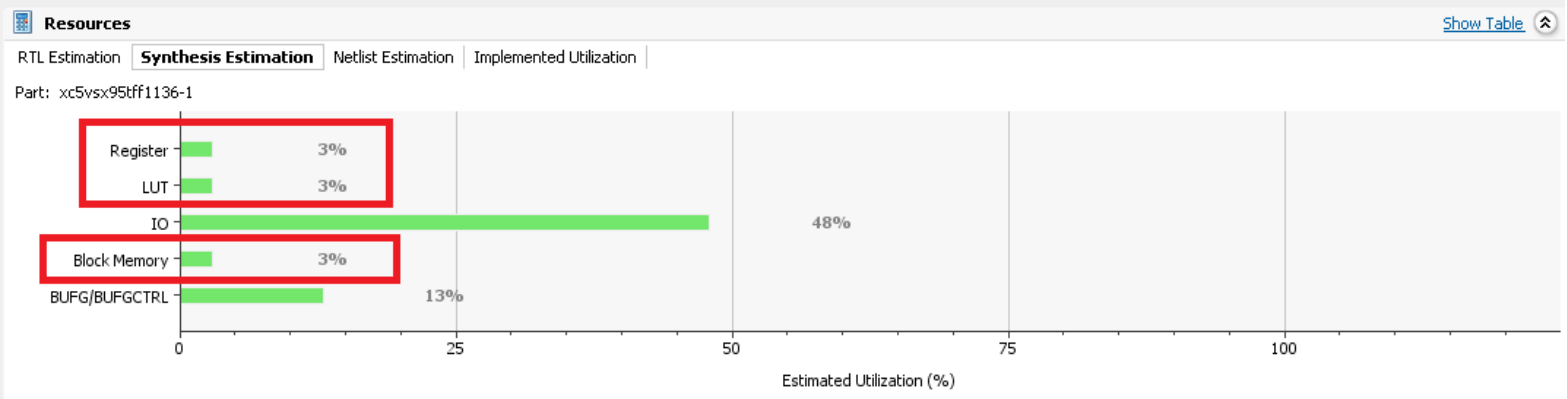
Fig. 2. Frame header

DSP development framework



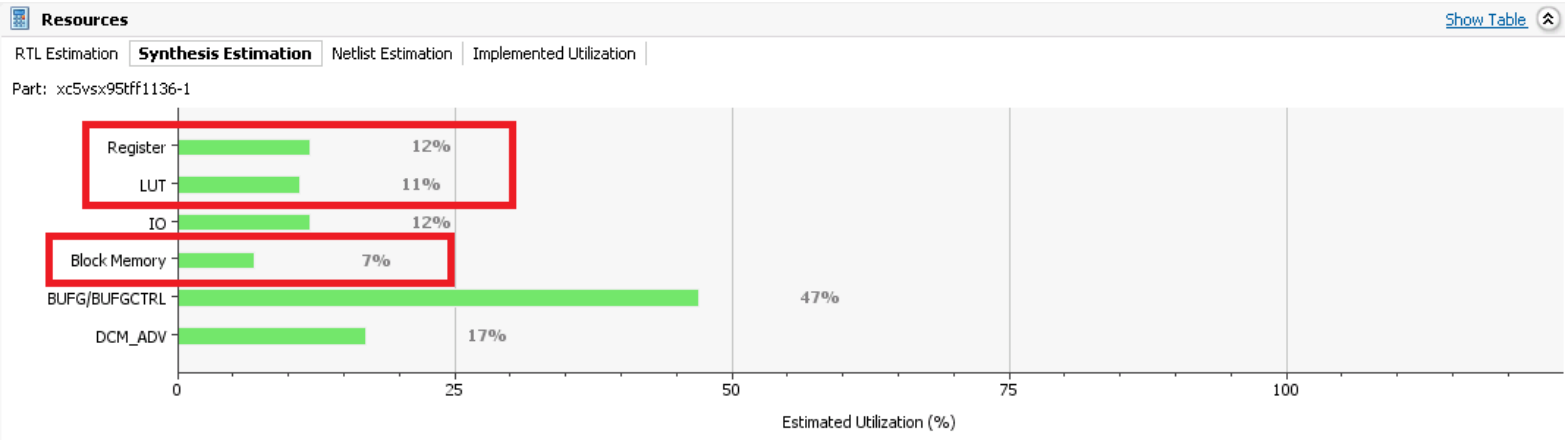
- Supports the communication protocol
- Easily expandable with additional algorithms with minimum programmer effort
- Low-level ’drivers’ already prepared
- Available reference designs - VHDL for uTC and C for DSP
- Data transmission via PCIe
- Data generation and readout in Matlab
- Console debug output from the DSP

Resource estimation - communication module

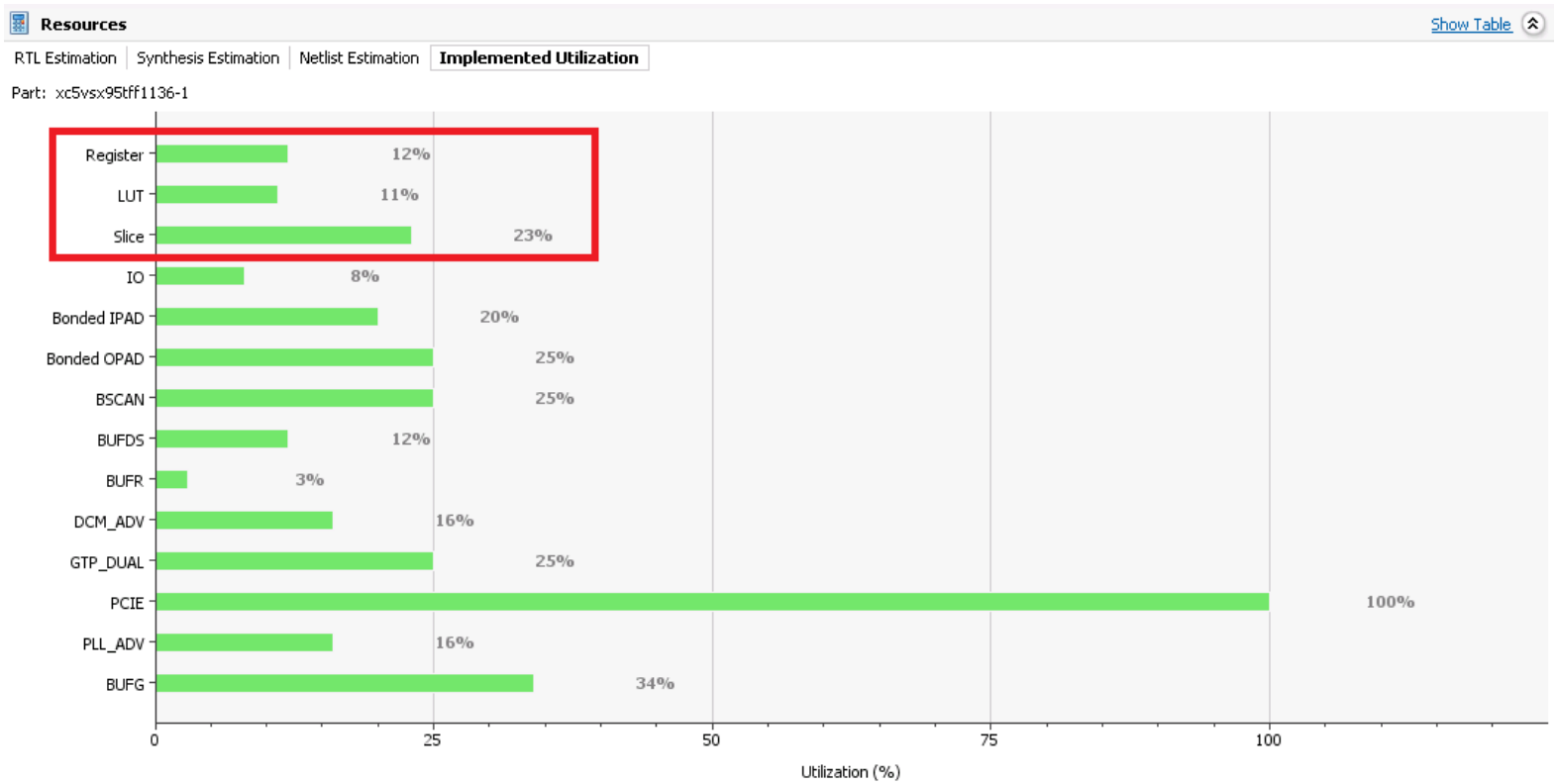


- Current uTC controller resource usage - 20%? 50%?

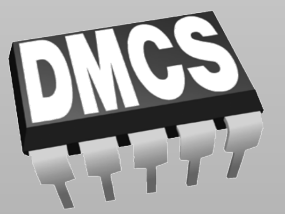
Resource estimation - reference project



Resource utilization - reference project

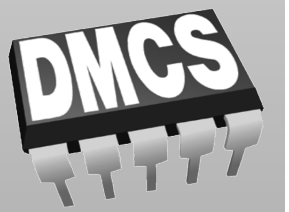


Outline

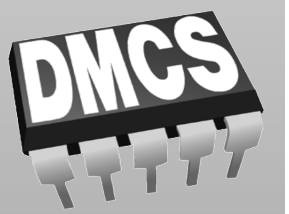


Timing issues

- Current DSP data clock = 100 MHz; works fine
- Attempt to go up to 200 MHz, but...
- ... DCM component switching issues - had to use PLL_ADV
- ... logic timing issues - requires careful floorplanning
- Tried for a couple of days but gave up - will make more sense after integration with LLRF controller



Further system development



- Integration with uTC LLRF controller - data format conversion: fixed $\rightarrow$ floating-point
- Introduction of 2nd or even 3rd transmission link - will result in 2x or 3x speed-up
- Implementation of useful algorithms in the DSP to aid the FPGA controller
- Resolution of timing issues (150 MHz, 200 MHz?) - even more throughput increase

Thank you for your attention

- Questions?
- Comments?