

A Streaming FFT on 3GSPS ADC Data using Core Libraries and DIME-C

Robin Bruce¹, Malachy Devlin²

¹Institute of System Level Integration, Alba Centre, Alba Campus, Livingston, EH54 7EG, UK, rbruce@sl-i-institute.ac.uk

²Nallatech, Boolean House, 1 Napier Park, Glasgow, UK, G68 0BH, m.devlin@nallatech.com

Introduction

The benefits of FPGA-based computing in the embedded context are now well established. For some embedded applications, size, weight and power constraints mean that FPGA-based systems are the only rational choice. FPGAs have powerful capabilities for integrating high-performance input and output devices that allow for the development of powerful but compact systems.^[1,2]

FPGA-based reconfigurable computers are generally regarded as being significantly more difficult to program than conventional microprocessor-based systems. Over the last few years a number of high-level language (HLL) compilers that target reconfigurable computers have been developed.^[3] More recently, the advantages of integrating core libraries into high-level languages have become more widely recognized.

The work presented here looks at the implementation of a 1024-point streaming fast Fourier transform (FFT) performed on a full data-rate pipe from the output of a 3 gigasamples per second (GSPS) analog-to-digital converter (ADC). The processing for this task is carried out by two Xilinx Virtex-4 FPGAs, an LX160 and an LX100. These FPGAs are situated on a BenADC-3G and a BenData-V4 module respectively. These modules are placed on a four-high BenNUEY-PCI-104 stack. Nallatech's DIME-C compiler is used to create the high-performance FFT processing structure by integrating firmware cores. The purpose of this project was not to produce an optimal hardware FFT core, see ^[5,6] for optimal hardware. Instead the purpose was to examine how a high-level tool might leverage predesigned cores to simplify embedded system development.

BenADC-3G

Advanced ADCs are set to enable an advance in embedded processing. Multi-gigasample ADCs allow digital receivers to be agile, supporting multiple applications over a broad range of frequencies and bandwidths. With digitized input, there is no longer a need for costly and bulky front-end processing and the removal of analog circuitry makes systems more reliable, repeatable and accurate. The BenADC-3G marries two 3 GSPS 8-bit ADC channels to a Virtex-4 FPGA (SX55 or LX160) on a DIME-II module.

DIME-C

Nallatech has developed the DIME-C tool both as a tool for application developers that use Nallatech reconfigurable computing platforms, as well as for use by Nallatech

researchers and engineers working on industrial and academic projects. DIME-C uses a syntax that is a subset of ANSI C. It will automatically seek to convert input C code into HDL that describes pipelined and parallel structures wherever possible. Research and development of DIME-C is ongoing. The feature set expands with time, and the conversion process that turns algorithms into HDL is improved to obtain the best performance possible for the hardware targeted. DIME-C offers the possibility of integrating low-level core libraries into the language to offer the best mix of high productivity, low resource use and high performance.

Core Libraries

High-level language compilers for FPGAs are often criticized for their suboptimal resource consumption and clock frequency when compared to expert HDL design. This limitation can be overcome through the use of core libraries. Hand-optimized HDL, with or without placement constraints, can be integrated into a high-level language compiler and used to create high-performance systems that make efficient use of the available resources. In suitable cases the core can be clocked at 2 or even 4 times the rate of the base HDL-generated logic.

Implementation

Figure 1 below shows the basic architecture for the FFT core. The BenADC3G firmware outputs sixteen 8-bit input data words at a rate of 187.5 MHz. It would be optimistic to expect DIME-C to be able to generate all its logic at this clock rate. Instead, the firmware that interfaces to the ADC is input to DIME-C as a library core at half that rate, meaning that 32 8-bit words are input every 93.75 MHz clock cycle. When in streaming mode, DIME-C reads in these 32 words on each clock cycle and stores them in locally declared arrays that utilize BRAM. From here, the data is fed into what, from the DIME-C compiler's perspective, appears to be 32 FFT cores, accessed in pairs. In reality these FFT cores are running at double the DIME-C clock rate at 187.5 MHz, and a system of FIFOs gives the illusion of two FFT cores operating at a slower rate.

¹This research is sponsored by Nallatech in conjunction with the Institute for System Level Integration.

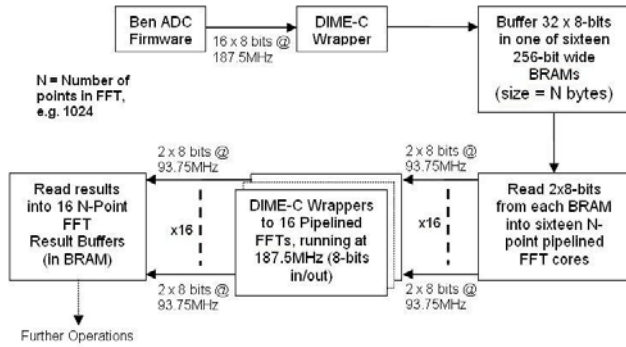


Figure 1: Parallel Streaming FFT Architecture

Xilinx Core Generator was used to generate an FFT core suitable for use. As can be seen from table 1 below, the FFT core consumed 1789 slices, 12 DSP48s and 9 RAMB16s. For the BenADC-3G there is a choice between an SX55 and an LX160 FPGA. The SX55 has 512 DSP48 blocks, each containing an 18-by-18 2's complement multiplier and an accumulator. The amount of general slice logic is limited however at 24,576 slices. The LX160 is at the opposite end of the spectrum, with an enormous amount of slice logic at 67,584 slices, but a limited 96 DSP48 blocks. Table 1 shows that in each of these devices only 8 FFT cores can be implemented. The LX160 is limited because of the DSP48s and the SX55 is slice limited. From Figure 1 we can see that the streaming FFT architecture requires 16 FFT cores, when both the SX55 and the LX160 can only accommodate 8 cores. It is recognized that the abundant slice logic of the LX160 would permit the construction of an alternative FFT core that used no DSP48s in its construction. However, the timescales of this project did not permit this optimization of resources. The LX160 was selected as the FPGA for the BenADC-3G in this case, as 8 FFT cores could be fitted on a single device with more than sufficient slices in reserve for implementing DIME-C logic, control logic and the necessary input/output firmware. To implement the other eight cores, a second FPGA is required. An LX100 is sufficient for this task and it is sited on a BenDataV4 module. Figure 2 shows the resultant system architecture. The two FPGAs carry out all major processing and are connected directly to each other over the local system interconnect.

Table 1: Resource Consumption of FFT Cores

No. of FFT Cores	1	8	16
Slices	1789	14281	28562
DSP48s	12	96	192
RAMB16s	9	72	144
% of LX160 Slices	2.6	21.2	42.4
% of LX160 DSP48s	12.5	100	200
% of SX55 Slices	7.3	58.4	116.8
% of SX55 DSP48s	2.3	18.4	36.8

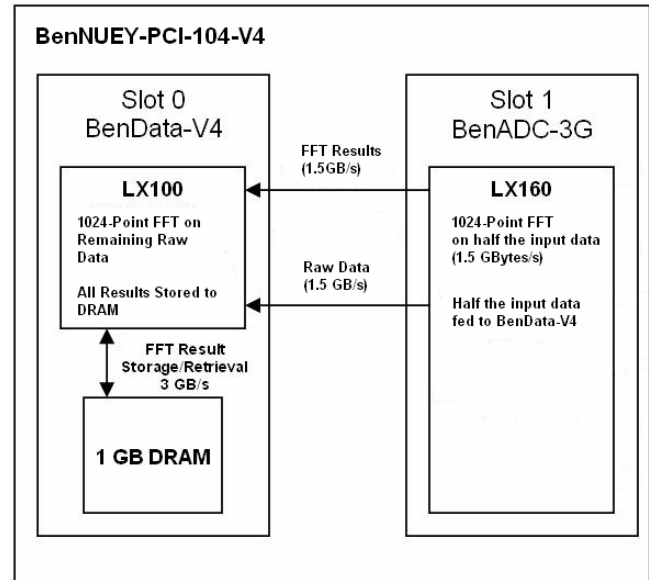


Figure 2: FPGA System Architecture

Conclusions

Using a high-level language for FPGAs with core libraries is an effective and productive manner in which to produce an embedded system with links to high-performance I/O. The majority of the resource consumption comes from the implementation of cores that are optimized and not from compiler-generated logic. The clock rate for the system is decided by the input data rate, and is not affected by the lower clock rate of the DIME-C-generated code.

References

- [1] GD Peterson, SL Drager, *Accelerating defense applications using high performance reconfigurable computing*, Government Microcircuit Applications and Critical Technology Conference, Tampa, FL, 31 March 3 April 2003.
- [2] Stephen Pearce / Mercury Computer Systems, Inc. Richard Jaenicke / Mercury Computer Systems, *Processing Challenges in Shrinking HPEC Systems into Small UAVs*, HPEC 2004, 28 September, MIT Lincoln Labs
- [3] Najjar, W. A. Bohm, W. Draper, B. A. Hammes, J. Rinker, R. Beveridge, J. R. Chawathe, M. Ross, C., *High-level language abstraction for reconfigurable computing*
- [4] Z Guo, A Mitra, W Najjar, *AUTOMATION OF IP CORE INTERFACE GENERATION FOR RECONFIGURABLE COMPUTING*, August 28 - 30, 2006, Madrid, Spain
- [5] Raymond J. Andraka, *Hybrid Floating Point Technique Yields 1.2 Giga-sample Per Second 32 to 2048 point Floating Point FFT in a single FPGA*, 19 - 21 September 2006, MIT Lincoln Labs
- [6] RF Engines <http://www.rfel.com/>