An Independent Evaluation
of
Floating-point DSP Energy Efficiency on Altera 28 nm FPGAs

By the staff of

Berkeley Design Technology, Inc.

February 2013

OVERVIEW

FPGAs are increasingly used as parallel processing engines for demanding digital signal processing (DSP) applications. Benchmark results show that on highly parallelizable workloads, FPGAs can achieve higher performance and superior cost/performance compared to digital signal processors and general-purpose CPUs. However, higher performance often comes with higher power consumption and lower energy efficiency, which can be problematic in embedded processing applications.

Altera recently introduced a floating-point design flow intended to streamline the process of implementing floating-point DSP algorithms on Altera FPGAs, and to enable those designs to achieve higher performance and resource usage efficiency than previously possible. In a previous white paper [1], BDTI performed an independent analysis to assess the performance of Altera FPGAs in demanding floating-point DSP applications and evaluated the effectiveness of Altera's floating-point DSP design flow.

Subsequently, BDTI performed an independent evaluation of the power consumption and energy efficiency of Altera FPGAs for demanding floating-point DSP applications. This white paper presents BDTI's findings from this follow-up evaluation.

© 2013 Berkeley Design Technology, Inc.

Contents

1. Introduction

Power consumption and energy efficiency are becoming increasingly important in the selection of high-performance embedded processors, because many systems must operate in confined spaces, in mobile environments, or on battery power. Since most of the power a processor uses is converted into heat, devices that consume less power require less cooling to avoid overheating. This in turn translates into smaller systems with smaller batteries. However, a low-power processor is not necessarily energy efficient. Energy consumption is determined by multiplying power consumption by time. In many cases, a lower-power processor also provides lower performance, and its lower power is cancelled out by the longer time required to complete a given task. Often, a low-power processor doesn't provide enough performance, whereas a high-performance processor may consume unacceptably high power.

Computationally demanding floating-point algorithms are becoming commonplace in embedded computing applications. Examples range from advanced military radar applications such as Space-Time Adaptive Processing (or STAP) to multiple-input multiple-output (MIMO) communications channel estimation in the fourth generation Long-Term Evolution (4G LTE) cellular standard. The complexity of these algorithms requires highly parallel processing techniques, and the environments in which they operate, such as confined spaces and small lightweight packages, require reduced power consumption. An example of such an environment is the mobile military arena including military unmanned aerial vehicles (UAVs). These applications demand not only low power consumption to minimize heat dissipation and maximize battery life, but also high computational performance to complete the given task in the shortest amount of time. A common metric used to assess energy efficiency in floating-point processors, which we use in this evaluation, is its performance measured in billions of floating-point operations per second, per Watt of power consumed or GFLOPS/W.

Processor vendors commonly publish peak GFLOPS numbers. Similarly, energy efficiency is often quoted in terms of peak GFLOPS divided by power consumption. These values assume that all floating-point functional units in the processor are running at the maximum clock frequency of the device. Such figures are usually very optimistic and do not reflect typical applications. In this analysis we have taken a different and a more realistic approach: We have measured both floating-point performance and power consumption based on implementations of two specific, complex algorithms. Altera recently introduced floating-point capability in the DSP Builder Advanced Blockset tool chain to simplify implementation of floating-point DSP algorithms on Altera FPGAs, while improving performance and efficiency of floating-point designs compared to traditional FPGA design techniques. In a previous white paper [1], BDTI evaluated the effectiveness of Altera's approach to floating-point design using the Quartus II software v12.0 tool chain and assessed the floating-point performance of Altera's 28 nm Stratix V and Arria V FPGAs. For that evaluation, we used two example applications, both designed to solve large sets of simultaneous linear equations using two different types of matrix decomposition: a multi-channel Cholesky matrix decomposition and a QR decomposition using the Gram-Schmidt process. These decompositions, combined with forward and backward substitutions, constitute a solution for the vector **x** in a simultaneous set of linear equations of the form **Ax** = **B**.

In this white paper, we evaluate the power consumption and energy efficiency of Stratix V and Arria V FPGAs using the two example applications from the previous white paper. As will be shown in Section 4, an Altera Stratix V FPGA can achieve over 6 GFLOPS/W while consuming 16 W of power at 99 GFLOPS, whereas Altera's Arria V FPGA can achieve over 7 GFLOPS/W at just over 9 W of power consumption at 65 GFLOPS. It's worthwhile to note that these are not peak performance figures, but rather the performance of realistic floating-point design examples.

Section 2 provides a brief background on the two floating-point example applications. Section 3 describes the power measurement methodology. Section 4 presents the results of the evaluation for the two example designs on two different Altera FPGAs: the high-end, medium-sized Stratix V 5SGSMD5K2F40C2N device and the low-power, midrange Arria V 5AGTFD7K3F40I3N device. Finally, Section 5 presents BDTI's conclusions.

2. Floating-Point Design Examples

Sets of linear equations of the form **Ax** = **b** arise in many fields, from advanced military radar applications such as STAP to various estimation problems in digital communications. Whether it is an optimization problem involving linear least squares, or MIMO communications channel estimation, the problem remains one of finding a numerical solution for the equation **Ax** = **b**. In addition to being computationally demanding, algorithms that solve these types of equations can suffer from numeric instability if sufficient dynamic range is not used. Therefore, efficient and accurate implementation of such algorithms is only practical in floating-point devices. For a general matrix **A** of size *m* by *n*, where *m* is the height of the matrix and *n* its width, QR decomposition may be used to solve for vector **x**. The algorithm decomposes **A** into an orthonormal matrix **Q** of size *m* by *n* and an upper triangular matrix **R** of size *n* by *n*. Since **Q** is orthonormal, $\mathbf{Q}^T\mathbf{Q} = \mathbf{I}$ and $\mathbf{Rx} = \mathbf{Q}^T\mathbf{b}$. Given that **R** is an upper triangular matrix, **x** can easily be solved by backward substitution without inverting the original matrix **A**. In the QR solver example in this white paper, we work with over-determined matrices with $m \geq n$ and we decompose matrix **A** using the Gram-Schmidt process.

When matrix **A** is Hermitian and positive definite, such as covariance matrices that arise in many applications, the Cholesky decomposition (which can be up to twice as efficient as QR decomposition) is commonly used. The algorithm decomposes **A** into a lower triangle **L** and its conjugate transpose **L***. Since **L** is a lower triangular matrix, the algorithm uses forward substitution to solve for **y** in **Ly** = **b**, followed by backward substitution to solve for **x** in **L*x** = **y**. Thus the algorithm indirectly finds the inverse of matrix **A** to solve for $\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$. The Cholesky solver example in this white paper has a multi-channel design, meaning that multiple matrices may be decomposed simultaneously.

Both solvers used for this evaluation are implemented using complex data and IEEE 754 single-precision floating-point arithmetic. Full detail on the two floating-point examples and their implementation on the two Altera FPGAs can be found in our previous white paper.

NOTATION AND DEFINITIONS

M Bold capital letter denotes a matrix.
z Bold small letter denotes a vector.

$\boldsymbol{L}^*$ The conjugate transpose of matrix L.

Hermitian Matrix A square matrix with complex entries that is equal to its own conjugate transpose. This is, the complex extension to a real symmetric matrix.

Positive Definite Matrix A Hermitian matrix **M** is positive definite if **z*Mz** > 0 for all non-zero complex vectors **z**. The quantity **z*Mz** is always real because **M** is a Hermitian matrix for the purposes of this paper.

Orthonormal Matrix A matrix **Q** is orthonormal if $\mathbf{Q}^T\mathbf{Q} = \mathbf{I}$ where **I** is the identity matrix.

Cholesky Decomposition A factorization of a Hermitian positive definite matrix **M** into a lower triangular matrix **L** and its conjugate transpose **L*** such that **M** = **LL***.

QR Decomposition A factorization of a matrix **M** of size *m* by *n* into an orthonormal matrix **Q** of size *m* by *n* and an upper triangular matrix **R** of size *n* by *n* such that **M** = **QR**.

F_{max} The maximum frequency of an FPGA design.

3. Power Measurement Methodology

We used two hardware platforms for this evaluation; the DSP Development Kit, Stratix V Edition, and the Arria V FPGA Development Kit. To use these platforms, developers download the DSP development kit installation software, unique to each hardware platform, and the USB-Blaster II driver from Altera's website. (This software is also available on a DVD which may be requested from Altera.) Included in the installation software download is an application package called the Board Test System. This environment provides a GUI interface to alter functional settings on the hardware development board and observe the results. The Board Test System setup communicates with the development board over a USB cable connected to the board's USB-Blaster II unit. The USB-Blaster controls the JTAG chain on the board.

The main DC power input on the board is stepped down via voltage regulators to supply

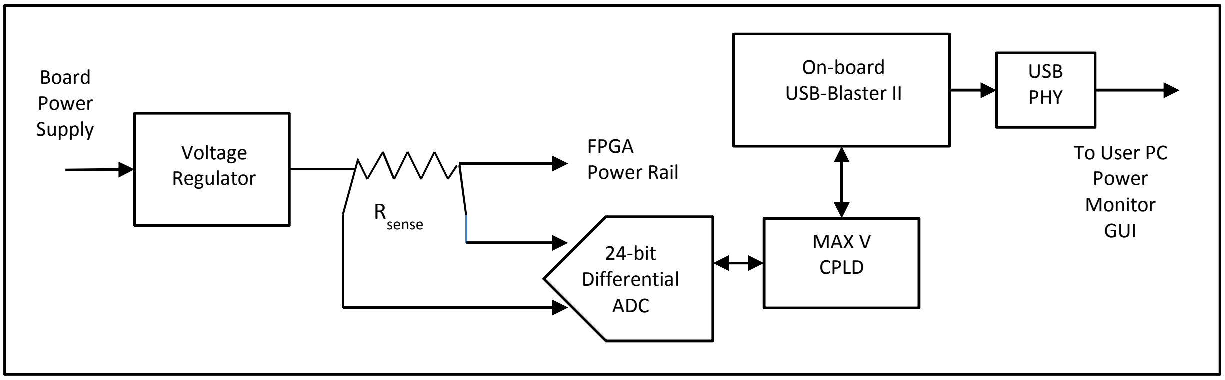


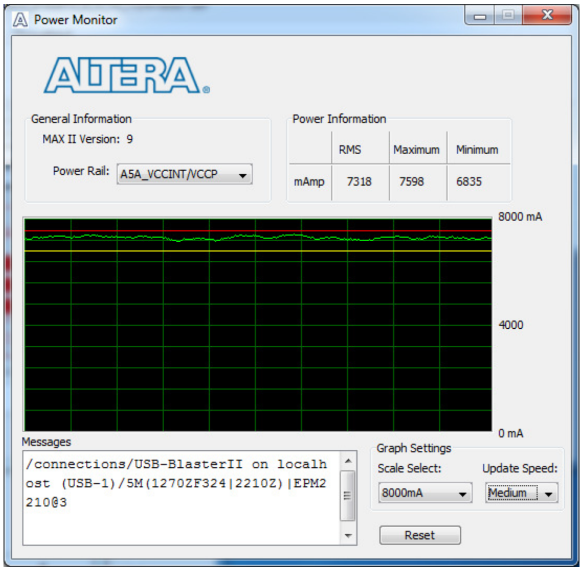
Figure 1. On-board power measurement circuitry for a single FPGA rail

power to the various power rails used by components on the board and on the FPGA. The FPGA power rails are split from the supply plane by low-value sense resistors. All FPGA power rails use a 0.003 Ω sense resistor, except for the FPGA core rail (VCCINT) which uses a 0.001 Ω resistor. Both types of resistors have an accuracy of 1%. 24-bit differential analog-to-digital converters (ADCs) are used to measure the voltage across the sense resistors. Each ADC communicates over a serial peripheral interface (SPI) bus with an Altera MAX V CPLD which acts as the on-board system controller for various functions such as FPGA configuration, power and temperature monitoring, and fan control. The MAX V CPLD is on the JTAG chain and thus communicates with the Board Test System application running on the user's PC. Both Stratix V and Arria V FPGA boards have essentially the same components other than the FPGA itself. Figure 1 shows the power measurement circuitry on the development boards for a single FPGA power rail. Each of the FPGA power rails has its own sense resistor.

For this evaluation, we used two applications; the Power Monitor and the Clock Control. Both part of the Board Test System environment, they may be run either through the Board Test System GUI or as stand-alone applications. In this evaluation, we chose to run these applications in their stand-alone mode.

The Clock Control application is used to set the frequency of the on-board programmable oscillators. On the Stratix V FPGA development board, we used the Si570 clock source, whereas on the Arria V FPGA development board we used the X7 clock source. For the Cholesky solver configurations, we set the oscillator frequency to $F_{max}/2$, since the Cholesky solver uses the on-chip PLL to double its input clock frequency. For the QR solver, we set the oscillator to the F_{max} for the configuration under evaluation. The F_{max} used for each configuration under test is indicated in Section 4, Table 1.

The Power Monitor application communicates with the on-board power monitoring circuitry, which measures and reports the current passing through the sense resistor for the various power rails on the board. We monitored the current passing through nine power rails for the Stratix V FPGA and seven power rails for the Arria V device. In each case, the floating-point application was running in continuous operation mode. Figure 2 shows the control GUI for the Power Monitor application for the Arria V FPGA Development Kit. The displayed RMS value for the current is the average of 16 values sampled over a period of 2 seconds. The sampling rate and the averaging period cannot be changed.



General Information

MAX II Version: 9

Power Rail: A5A_VCCINT/VCCP

Power Information

	RMS	Maximum	Minimum
mAmp	7318	7598	6835

Messages

```
/connections/USB-BlasterII on localh
ost (USB-1)/5M(1270ZF324|2210Z)|EPM2
210@3
```

Graph Settings

Scale Select: 8000mA

Update Speed: Medium

Figure 2. Power Monitor GUI

© 2013 Berkeley Design Technology, Inc.

However, the update rate of the GUI and the graphical display may be controlled by the user. The MAX and MIN values displayed are the absolute maximum and minimum RMS values encountered during an entire run. The precision of the display is 1 mA.

To get an accurate current measurement for each configuration of each floating-point example, we started the application in continuous mode and waited until the FPGA reached its operating temperature before recording the current values. The displayed RMS current value rises initially as the device temperature rises, and stabilizes around its long-term average. This process took up to seven minutes in some cases. We then monitored and recorded the RMS current values and averaged them over a period of about two minutes. Although the current values were relatively stable over this period, averaging still helps in smoothing out small variations. In order to get an accurate figure for the displayed current value, we independently measured the voltage drop across the VCCINT rail (FPGA core power rail) sense resistor for a few of the designs and compared the calculated current values to the corresponding displayed values. Our estimated error margin for the power monitor display is within ± 1%. To calculate the power consumed on each rail, we multiplied the averaged RMS current by the power rail voltage. The complete list of the power rails that we monitored, and their corresponding voltages are shown in Table A.1 of the Appendix.

4. Power Results

This section presents BDTI's independent evaluation of the power consumption and energy efficiency for two Altera 28 nm FPGAs: the high-end, medium-sized Stratix V 5SGSMD5K2F40C2N device and the low-power, midrange Arria V 5AGTFD7K3F40I3N device. The Stratix V FPGA used in this evaluation, features 345.2K adaptive look-up tables (ALUTs), 1,590 27×27-bit variable-precision multipliers, and 2,014 M20K memory blocks. The Arria V FPGA features 380.4K ALUTs, 1,156 27×27-bit variable-precision multipliers, and 2,414 M10K memory blocks. The power consumption of both FPGAs was measured while running the complex-data, single-precision IEEE 754 floating-point Cholesky and QR solver examples.

Table 1 presents the energy efficiency achieved on Altera's Stratix V and Arria V FPGAs in units of GFLOPS/W (last column) when running each of the two floating-point examples in a continuous mode. The throughput, F_{max}, and performance columns in Table 1 are results repeated here from BDTI's performance evaluation for the same FPGAs described in [1]. The equations used to calculate the number of real-data floating-point operations per second for the Cholesky and QR solver design examples are $4n^3/3 + 12n^2$ and $8mn^2 + 6.5n^2 + mn$, respectively.

The power figures presented in Table 1 are the total measured power consumption for each of the Cholesky and QR solver configurations on both Stratix V and Arria V FPGAs. The energy

Example	Device	Configuration (Channel Size/ Matrix Size/ Vector Size)	Throughput (kMatrices/ sec)	Fmax (MHz)	Performance (GFLOPS)	Total Power[1] (W)	GFLOPS/ W
Cholesky	Stratix V	1 / 360×360 / 90	1.43	189	91	16	5.7
		20 / 60×60 / 60	118.35	234	39	15	2.6
		64 / 30×30 / 30	544.28	288	26	10	2.5
	Arria V	6 / 90×90 / 45	35.22	197	38	9.1	4.2
		64 / 30×30 / 30	349.62	184	16	7.1	2.3
QR	Stratix V	1 / 400×400 / 100	0.315	203	162	26	6.2
		1 / 200×100 / 100	8.76	207	141	23	6.1
		1 / 200×100 / 50	6.17	260	99	16	6.2
		1 / 100×50 / 50	32.82	259	66	13	5.1
	Arria V	1 / 200×100 / 50	4.05	171	65	9.1	7.1
		1 / 100×50 / 50	21.54	170	44	8.1	5.4

Table 1. Power efficiency of Stratix V and Arria V FPGAs running Cholesky and QR solvers.

[1] Power values have an error margin of ±1 %.

efficiency in GFLOPS/W presented in Table 1 is calculated by dividing the performance value by the total measured power consumption for each case.

The total power consumption for each configuration includes the sum of power consumed on all the power rails of each FPGA. Although in many applications, including the examples we used in this white paper, some sections of the FPGA, such as the transceivers, are not actively used, they nevertheless contribute to static power consumption and we have included these as part of the reported totals in Table 1. In these examples, the FPGA core consumption ranged from 82% to 92% of the reported total power values.

Tables A.2 through A.5 in the Appendix present the RMS current measurements and the calculated power consumption per power rail for the two floating-point design examples on both Stratix V and Arria V FPGAs.

The two floating-point design examples we evaluated in this white paper were compiled to maximize performance, i.e., F_{max}. Designers using the Quartus II software v12.0 tool chain have several options to reduce power consumption in their design through power-driven compilation, clock power management, and optimized memory clocking. The Quartus II Handbook version 12.1, volume 2 chapter 14, discusses optimization techniques related to power consumption, and may be downloaded from Altera's website [2].

5. Conclusions

In this white paper, we evaluated the energy efficiency and the power consumption of two 28 nm Altera devices: the high-end, medium-sized Stratix V FPGA and the low-power, midrange Arria V FPGA. The energy efficiency was evaluated on two design examples, the Cholesky and the QR solvers, implemented using single-precision, complex-data IEEE 754 floating-point numbers. Both examples were designed and implemented using Altera's Quartus II software v12.0 tool chain, and presented in detail in our previous white paper.

Our evaluation shows that the Altera Stratix V FPGA can achieve high computational performance executing complex floating-point applications with power consumption low enough to enable use in many power-sensitive embedded systems. The largest floating-point example that we evaluated was a 400 × 400 element QR solver on the Stratix V FPGA. Running at 203 MHz and processing 162 GFLOPS, the device achieved a power efficiency of 6.2 GFLOPS/W while consuming 26 W. Comparing the two Altera FPGAs running identical floating-point design configurations, we observed that although the Arria V FPGA has lower performance and hence lower power consumption, nevertheless its energy efficiency is comparable to that of the Stratix V FPGA. Moreover, for similar computational performance (GFLOPS), we observed that the Arria V FPGA achieves lower power consumption and higher energy efficiency than the Stratix V FPGA. These two observations indicate that the Arria V FPGA has both lower static and lower dynamic power consumption than the Stratix V device.

Finally, it must be noted that the performance and the energy efficiency numbers for these two Altera FPGAs presented in this white paper are for specific design examples and do not represent peak values under specialized circumstances. To enable valid comparisons with other platforms, the same algorithms should be implemented on those platforms and their energy efficiency and power consumption measured.

6. References

[1] Berkeley Design Technology, Inc., October 2012. "An Independent Analysis of Floating-point DSP Design Flow and Performance on Altera 28-nm FPGAs". Available for download at http://www.altera.com/literature/wp/wp-01166-bdti-altera-floating-point-dsp.pdf.

[2] "Quartus II Handbook version 12.1, volume 2" available for download from Altera's website at http://www.altera.com/literature/hb/qts/qts_qii5v2.pdf.

Appendix

Current measurements have a margin of error of ±1 %.

Table A.1. Monitored power rails and their voltages

Stratix V FPGA		
Power Rails	**Voltage (V)**	**Function**
VCCINT	0.90	FPGA core
XCVR_GXB	1.0	High-speed transceiver
VCCIO_HSMB	1.2	VCC I/O
VCCPD/PGM	2.5	I/O pre-driver, programming
VCC_1.5	1.5	PLL, transceiver buffers
VCCIO_1.8	1.8	1.8 V I/O
VCCIO_2.5	2.5	2.5 V I/O
VCCIO_1.5	1.5	1.5 V I/O
VCCA_GXB	3.0	XCVR analog power

Arria V FPGA		
Power Rails	**Voltage (V)**	**Function**
VCCINT/VCCP	1.1	FPGA core
VCCD_PLL	1.5	Digital portion of PLL
VCCIO_1.5V	1.5	1.5 V I/O
VCCIO_1.8V	1.8	1.8 V I/O
VCCA	2.5	Analog power for PLL
VCCPD/PGM	2.5	I/O pre-driver, programming
XCVR_GXB	1.2	High-speed transceiver

Table A.2. Power consumption for the Cholesky solver on the Stratix V FPGA

Number of channels = 1 **Matrix size = 360 x 360** **Dot product vector size = 90** **F_{max} = 189 MHz**		
Power Rail	**Msr'd Current (mA)**	**Power (W)**
VCCINT	15530	14
XCVR_GXB	300	0.30
VCCIO_HSMB	2	0.002
VCCPD/PGM	90	0.23
VCC_1.5	496	0.74
VCCIO_1.8	0	0
VCCIO_2.5	0	0
VCCIO_1.5	0	0
VCCA_GXB	238	0.71
Total Power Consumption	16	
% Consumed by Core	88%	
GFLOPS/W	5.7	

Number of channels = 20 **Matrix size = 60 x 60** **Dot product vector size = 60** **F_{max} = 234 MHz**		
Power Rail	**Msr'd Current (mA)**	**Power (W)**
VCCINT	14474	13
XCVR_GXB	299	0.30
VCCIO_HSMB	2	0.002
VCCPD/PGM	90	0.23
VCC_1.5	488	0.73
VCCIO_1.8	0	0
VCCIO_2.5	0	0
VCCIO_1.5	0	0
VCCA_GXB	238	0.71
Total Power Consumption	15	
% Consumed by Core	87%	
GFLOPS/W	2.6	

Number of channels = 64 **Matrix size = 30 x 30** **Dot product vector size = 30** **F_{max} = 288 MHz**		
Power Rail	**Msr'd Current (mA)**	**Power (W)**
VCCINT	9368	8.4
XCVR_GXB	271	0.27
VCCIO_HSMB	2	0.002
VCCPD/PGM	91	0.23
VCC_1.5	455	0.68
VCCIO_1.8	0	0
VCCIO_2.5	0	0
VCCIO_1.5	0	0
VCCA_GXB	238	0.71
Total Power Consumption	10	
% Consumed by Core	82%	
GFLOPS/W	2.5	

Table A.3 Power consumption for the Cholesky solver on the Arria V FPGA

Number of channels = 6 Matrix size = 90 x 90 Dot product vector size = 45 F_{max} = 197 MHz		
Power Rail	Measured Current (mA)	Power (W)
VCCINT/VCCP	7341	8.1
VCCD_PLL	6	0.009
VCCIO_1.5 V	17	0.026
VCCIO_1.8 V	17	0.031
VCCA	336	0.84
VCCPD/PGM	13	0.033
XCVR_GXB	29	0.035
Total Power Consumption	9.1	
% Consumed by Core	89%	
GFLOPS/W	4.2	

Number of channels = 64 Matrix size = 30 x 30 Dot product vector size = 30 F_{max} = 184 MHz		
Power Rail	Measured Current (mA)	Power (W)
VCCINT/VCCP	5531	6.1
VCCD_PLL	6	0.009
VCCIO_1.5 V	17	0.026
VCCIO_1.8 V	15	0.027
VCCA	336	0.84
VCCPD/PGM	15	0.038
XCVR_GXB	25	0.030
Total Power Consumption	7.1	
% Consumed by Core	86%	
GFLOPS/W	2.3	

Table A.4a. Power consumption for the QR solver on the Stratix V FPGA

Number of channels = 1 Matrix size = 400 x 400 Dot product vector size = 100 F_{max} = 203 MHz		
Power Rail	Measured Current (mA)	Power (W)
VCCINT	26258	24
XCVR_GXB	359	0.36
VCCIO_HSMB	3	0.004
VCCPD/PGM	86	0.22
VCC_1.5	596	0.89
VCCIO_1.8	0	0
VCCIO_2.5	2	0.005
VCCIO_1.5	0	0
VCCA_GXB	240	0.72
Total Power Consumption	26	
% Consumed by Core	92%	
GFLOPS/W	6.2	

Number of channels = 1 Matrix size = 200 x 100 Dot product vector size = 100 F_{max} = 207 MHz		
Power Rail	Measured Current (mA)	Power (W)
VCCINT	22882	21
XCVR_GXB	337	0.33
VCCIO_HSMB	3	0.004
VCCPD/PGM	86	0.22
VCC_1.5	560	0.84
VCCIO_1.8	0	0
VCCIO_2.5	2	0.005
VCCIO_1.5	0	0
VCCA_GXB	240	0.72
Total Power Consumption	23	
% Consumed by Core	91%	
GFLOPS/W	6.1	

Table A.4b. Power consumption for the QR solver on the Stratix V FPGA

Number of channels = 1 Matrix size = 200 x 100 Dot product vector size = 50 F_{max} = 260 MHz		
Power Rail	**Measured Current (mA)**	**Power (W)**
VCCINT	15470	14
XCVR_GXB	300	0.30
VCCIO_HSMB	3	0.004
VCCPD/PGM	86	0.22
VCC_1.5	490	0.74
VCCIO_1.8	0	0
VCCIO_2.5	2	0.005
VCCIO_1.5	0	0
VCCA_GXB	238	0.71
Total Power Consumption	16	
% Consumed by Core	88%	
GFLOPS/W	6.2	

Number of channels = 1 Matrix size = 100 x 50 Dot product vector size = 50 F_{max} = 259 MHz		
Power Rail	**Measured Current (mA)**	**Power (W)**
VCCINT	12131	11
XCVR_GXB	295	0.30
VCCIO_HSMB	3	0.004
VCCPD/PGM	86	0.22
VCC_1.5	481	0.72
VCCIO_1.8	0	0
VCCIO_2.5	2	0.005
VCCIO_1.5	0	0
VCCA_GXB	239	0.72
Total Power Consumption	13	
% Consumed by Core	85%	
GFLOPS/W	5.1	

Table A.5. Power consumption for the QR solver on the Arria V FPGA

Number of channels = 1 Matrix size = 200 x 100 Dot product vector size = 50 F_{max} = 171 MHz		
Power Rail	**Measured Current (mA)**	**Power (W)**
VCCINT/VCCP	7467	8.2
VCCD_PLL	5	0.008
VCCIO_1.5 V	15	0.023
VCCIO_1.8 V	15	0.027
VCCA	307	0.77
VCCPD/PGM	14	0.035
XCVR_GXB	23	0.028
Total Power Consumption	9.1	
% Consumed by Core	90%	
GFLOPS/W	7.1	

Number of channels = 1 Matrix size = 100 x 50 Dot product vector size = 50 F_{max} = 170 MHz		
Power Rail	**Measured Current (mA)**	**Power (W)**
VCCINT/VCCP	6443	7.1
VCCD_PLL	3	0.005
VCCIO_1.5 V	14	0.021
VCCIO_1.8 V	10	0.018
VCCA	344	0.86
VCCPD/PGM	14	0.035
XCVR_GXB	22	0.026
Total Power Consumption	8.1	
% Consumed by Core	88%	
GFLOPS/W	5.4	