Interfacing Reversible Pass-Transistor CMOS Chips with Conventional Restoring CMOS Circuits

Stéphane Burignat[1], Michael Kirkedal Thomsen[2], Michał Klimczak[1], Mariusz Olczak[1], and Alexis De Vos[1,3]

[1] Universiteit Gent, Vakgroep Elektronica en Informatiesystemen, Sint Pietersnieuwstraat 41, 9000 Gent, Belgium
[2] University of Copenhagen, Department of Computer Science, DIKU, Universitetsparken 1, 2100 Copenhagen, Denmark
[3] Imec v.z.w., Kapeldreef 75, 3001 Leuven, Belgium
research@burignat.eu, shapper@diku.dk, alex@elis.ugent.be

Abstract. Recent progress on the prototyping of reversible digital circuits, have shown that *adiabatic reversible dual-line pass-transistor logic* can be used for special purpose applications in reversible computation. This, however, raises new issues regarding the compatibility between this adiabatic logic implementation and conventional CMOS logic. The greatest difficulty is brought by the difference in signal shape used by these two logic families. Whereas standard switching CMOS circuits are operated by rectangular pulses, dual-line pass-transistor reversible circuits are controlled by triangular or trapezoidal signals to ensure adiabatic switching of the transistors. This work proposes a simple technical solution that allows interfacing reversible pass-transistor logic with conventional CMOS logic, represented here by an FPGA embedded in a commercial Xilinx Spartan-3E board. All proposed solutions have successfully been tested, which enables the FPGA to perform calculations directly on a reversible chip.

1 Introduction

Conventional computing is based on logically irreversible elementary operations, which leads to information destruction and an increase in entropy [1,2]. Eliminating the irreversibility of computations *is* possible. It was first shown by Bennett [3], whome theoretically enabled calculation processes with zero energy dissipation at finite speed. At the circuit level, this is only possible using reversible logical "machines" that perform thermodynamically reversible computations. Such "machines" have been conceptually proposed by Fredkin and Toffoli [4] and later extended for a quantum mechanics based computational model by *eg.* Feynman [5]. CMOS implementations of these logic operators were proposed by De Vos *et al.* [6] using dual-line pass-transistor (DLPT) circuits.

A. De Vos and R. Wille (Eds.): RC 2011, LNCS 7165, pp. 112–122, 2012.

This CMOS dual-line pass-transistor technology is demonstrated to be energetically efficient; results of simulations show energy consumption down to only 5 % of conventional CMOS [7]. In effect, biases are only used for substrate and wells polarization, while the necessary energy used for computation is brought only by the input signals. Moreover, the complementary signals ensure that the charge flowing in the circuits can be reused at next step of the computation, instead of being thrown away as in conventional restoring electronics. By opposition, restoring electronics makes use of biases to perform the computation and restore the signal at the same time, throwing away a large amount of energy at each clock cycle.

Whereas the classical digital CMOS switching technology uses square-wave waveforms to define logic "0" and logic "1", DLPT with adiabatic switching necessitates two complementary signals. Furthermore, it is mandatory that the two signals change gradually to ensure accurate adiabatic switching of the circuits [8] (see Fig. 1(a)). These signals are often triangular or trapezoidal waveforms. We refer to these signals by the name *adiabatic signals*.

Another advantage of adiabatic signals, is that these also avoid calculation errors, possibly caused by delays appearing between signals when classical rectangular shape signals involve steep transition slopes. The undesired pulses are then filtered when the smooth triangular slope amplitude is lower than the threshold voltage of the pass-transistor gates, thereby lowering the undesired artifacts [8] (output pulse shown in Fig. 1(b)).

We expect that, in its first "commercial" applications, reversible electronics will appear in ASICs and will be embedded in an environment of conventional CMOS circuits. Interfacing irreversible restoring logic and adiabatic logic has been proposed by Amirante *et al.* [9]. But their implementation is based on a different adiabatic logic family, by which their results can not be transfered directly to the reversible DLPT logic we are using. In this work, we propose a solution for interfacing a commercial FPGA, first with a reversible CMOS reversible binary adder [8,10,11] and then later with a reversible ALU [12]. First, in Sect. 2 we present the possibilities and the limitations caused by the Xilinx Spartan 3E FPGA [13]. Then, in Sect. 3 the proposed technical electronic solutions are detailed. Finally in Sect. 4 we conclude.

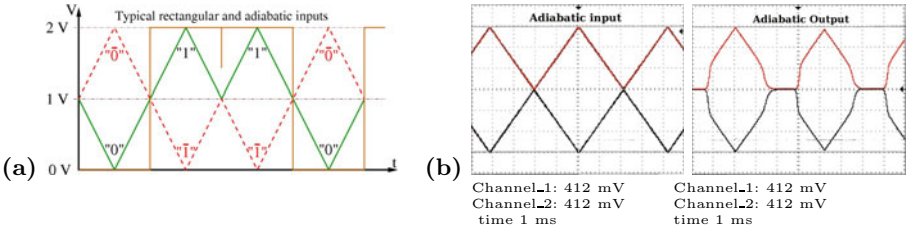

Fig. 1. (a) Typical square-wave and adiabatic dual-line inputs. (b) Experimental measurements of one dual input and one dual output using the a reversible binary adder in reverse calculation.

2 FPGA Capabilities and Limitations

Modern FPGA boards are much more than a simple chip of programmable hardware with a small memory and a parallel port to load the program. Even smaller FPGA starter-boards include a wide range of connector devices, which can be interfaced directly from the FPGA in a more or less simple way.

For the purpose of connecting the reversible chip with a digital circuit, it suffices to use an FPGA with a low number of logic cells. Therefore, the cheap (less than 200 euros) Xilinx Spartan®-3E FPGA Starter Kit Board from Digilent is used for this proof-of-concept. This board has a large number of input and output devices, a good manual, and, as it is a popular starter board, it is easy to find example implementations of device interfaces in both VHDL and Verilog. The on-board devices we are using are

- the clock generator: the standard clock source with a frequency of 50 MHz,
- the DAC: the digital-to-analog convertor, used to generate the adiabatic signals for the reversible chip inputs,
- the FX2 100-pin expansion connector for connecting the digital inputs and outputs to the connector board,
- the LCD display, used to show the results of the calculations, and
- the buttons, switches and the knob for controlling the board while executing.

The detailed use of each device will be described in the following sections.

2.1 Signal Generation

The adiabatic signal is normally generated with an accurate full-wave rectifier connected to a waveform generator [8]. This setup ensures a powerful and precise signal that is necessary for performing accurate measurements, but the setup is too big to be used in computers and especially small devices. This work only focuses on interfacing the reversible chip with classical CMOS circuits. We therefore need to ensure that the functionality of the reversible chip is correct, while the full computing circuit still gains the benefits of the reversible chip and the adiabatic switching. We use the DAC (included on the FPGA board) to generate the adiabatic signals. The DAC (presented in Fig. 2) is a *Lineal Technology LTC2624*. It has four outgoing pins, that can be controlled individually or all at once. By default, two of the outputs (pins A and B) have a reference[1] voltage of 3.3 V, while the two other pins (pins C and D) have a reference voltage of 2.5 V. The DAC output voltage is linearly controlled through a *Serial Peripheral Interface* (SPI). The accessible voltage values range between 0 V and 3.3 V or 2.5 V depending on the chosen pin and a value represented by a 12 bit number. The given voltage accuracy is 5 %.

[1] The default reference voltages of the DAC can be changed, but this requires desoldering of two resistors, and direct interfacing of two pins. Therefore, we have chosen to use the default settings.

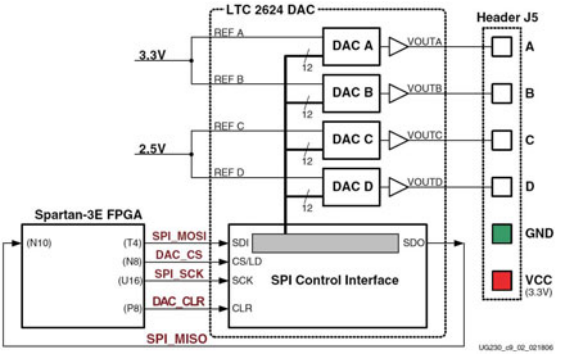

Fig. 2. Schematic of the digital to analog converter. *Figure adapted from [13].*

Fig. 3. Generation of the trapezoid signal for adiabatic switching

Often, a triangular waveform is used for adiabatic switching. But, as we will see in Sect. 2.2, it can be beneficial to use a trapezoidal shape instead. The steps to generate both these waveforms are illustrated in Fig. 3. First, we make a saw-tooth by incrementing a value in each clock cycle *(1)*. By doing the incrementing modulo some maximum number[2] we have the sudden value drop that is a characteristic of the saw-tooth waveform. Second step, is to make a triangular waveform and this is done by subtracting the most significant half of the saw-tooth waveform from the least significant half *(2)*. Finally, the trapezoidal waveform is obtained by removing the tops from the triangular waveform *(3)*. This is a simple greater-than check and the upper limit can be used to control the slope of the trapezoid waveform.

The period of the adiabatic signals are, of course, limited by the DAC. For each change of voltage level by the DAC, we need to send a 32-bit value through the SPI. Four FPGA clock cycles are needed in order to send each bit. As about 50 points are necessary to generate a well-defined adiabatic waveform for each of the two DAC outputs, and because the FPGA runs at 50 MHz, then the maximum reachable frequency for the adiabatic signal is about 4 kHz. This is not much, but it is fast enough for this proof-of-concept, which aims to show the possibility to interface the two technologies.

[2] The modulo 2^n operation is done automatically when using an n-bit adder circuit.

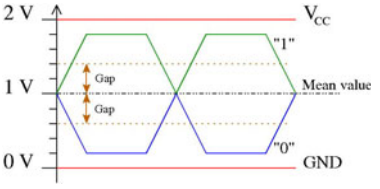

Fig. 4. Analog signals generated from the DAC

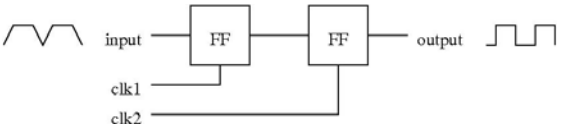

Fig. 5. Two-level memory interfacing the adiabatic input and the digital output signals

2.2 Interfacing and Timing with Internal Clock and Memory

The adiabatic waveforms are periodical and, therefore, have a build-in clock period. Moreover, both the digital circuits and the reversible chip must communicate at the same frequency[3]. Setting the phases between the signals from the FPGA and the adiabatic signal is not obvious when we want the circuits to run in a correct adiabatic way. We need to pay particular attention to the interfacing between the adiabatic signal and the conventional memory.

For the adiabatic signal a logic value is well-defined, either if its amplitude voltage exceeds the mean value by a quarter of the total signal range, or if it reduces by the same voltage gap (illustrated in Fig. 4). If trapezoidal waveforms are used then the signals are best defined during the constant plateau situated between the raising and falling transitions. If triangular waveforms are used, then the signal is best defined during a shorter time ranging from half the raising time to half the falling time of the triangular waveforms.

This is in contrast to digital circuits, where the well-defined logic values are situated just before the clock ticks and the logic value is changed. It is therefore not possible to make an interface with a single memory element and an approach using a two-level memory as in Fig. 5 must be applied.

The idea is that the first flip-flop reads the adiabatic signal at the clearest logic value, while the second flip-flop is updated at the change of the adiabatic signal. This interfacing works best if we are using a trapezoidal waveform, as these can be interpreted at their constant plateau as either digital logic “1” or a digital logic “0”.

To have an optimal timing between the two flip-flops, we must therefore have two separate internal clock signals. Using more clock signals to control energy-efficient circuits with adiabatic switching, is not a new idea. The *split-level charge*

[3] Other parts of the FPGA, such as the generation of the adiabatic signal and control of devices, will run with the standard 50 MHz clock.

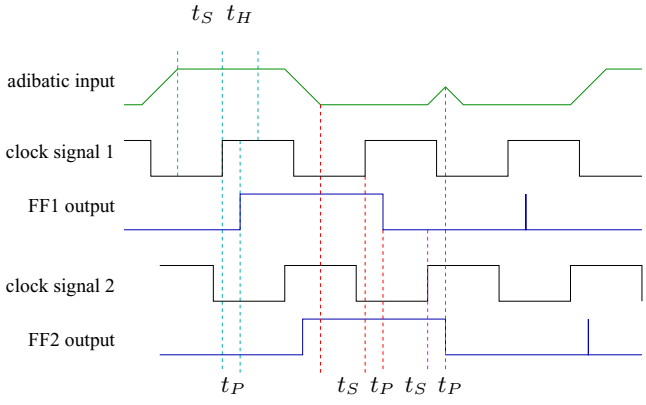

Fig. 6. Detailed timing of the interface between adiabatic signal and memory. t_S is the setup time, t_H the hold time, and t_P the propagation time of the memory.

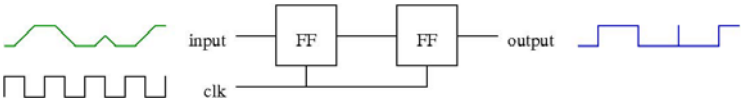

Fig. 7. Simplified interfacing with a single clock signal that is in-phase with the period of the adiabatic signal. The first flip-flop is updated at rising clock-edge whereas the second one at falling clock-edge.

recovery logic (SCRL) [14], which was used to implement the Pendulum processor [15], is controlled with up to 7 different clock signals. Both in SCRL and in our approach timing of the clock signals are essential. Therefore, we must consider the setup time, hold time, and propagation delay of the memory.

Fig. 6 illustrates a detailed timing diagram. This diagram does not consider signal propagation in the combinatorial circuits that has to be added to the minimum clock period. The figure is only intended to show that the trapezoidal adiabatic signal can uphold the timing constraint of the digital memory, while still ensuring adiabatic switching.

In the current setup, the maximum frequency of the adiabatic waveform is many times larger than the time constraints of the FPGA memory and the detailed timing is, therefore, not necessary. It is acceptable to make a simplified implementation with a single clock signal that is in-phase with the period of the adiabatic signal (shown in Fig. 7). The first flip-flop is updated at the rising clock-edge, where there is a clear adiabatic signal while the second flip-flop is updated at the falling clock-edge, where the adiabatic signal is switching.

2.3 I/O Programming

To transfer the input and output values for reversible circuits from/to the FPGA, a 100-pin Hirose FX2 connector is used. Most of these 100 pins have a special

purpose, leaving only 38 I/O pins and 5 input-only pins. As the connecting signals are digital and not complementary dual-line signals, we can control up to 21 inputs and 21 outputs to the reversible chip. The last pin from the connector will be used to control the execution direction on the chip, such that the FPGA can exploit the chip's reversibility, and not only the energy savings.

3 The Interface Board

In the design of the interface board, four electronic challenges have to be addressed.

- First, to convert classical rectangular signals into dual adiabatic signals, in order to transmit data to the inputs of the reversible chip.
- Second, the opposite, *i.e.* to transmit the computed information back to the FPGA in the form of a classical rectangular signal, starting from the dual-line adiabatic outputs of the chip.
- Third, to exploit the reversibility of the chip by making computation in both directions; the same chip may perform both the do-calculation (addition in this paper) and the undo-calculation (subtraction).
- Fourth, to synchronize the signals at the output interface.

3.1 Bringing the FPGA Commands to the Reversible Inputs

The conversion from classical square-wave signals to dual-line adiabatic signals (shown in Fig. 8a) can be done using a double $1 \rightarrow 2$ pass-transistor multiplexer: *e.g.* the commercial MUX SN74CBTLV3257. As the multiplexer is made of pass-transistor gates, it can be used in both directions.

The command inputs to the multiplexer are the enable $\overline{\text{oe}}$ input and the selection bit **s**. When $\overline{\text{oe}}$ is "1", the circuit is placed in high-impedance such that no input signal can be transmitted to output. When $\overline{\text{oe}}$ is "0", the selection bit s will act as the control to a switch and redirect the input signal to one of the two outputs. By applying the logical signal coming from the FPGA as the selection bit s, it is possible to connect the input to either output A if $s =$ "0" or else output B if $s =$ "1". Then, by adequately connecting the four outputs to the adiabatic logic "1" and logic "0" for the two first outputs and then logic "0" and logic "1" for the two next, we can obtain the mandatory dual-line adiabatic signals: logic "0" if $s = 0$ and logic "1" if $s = 1$.

The advantage of this solution is that it is already implemented with pass-transistor gates (Fig. 8b). In total, three pass-transistor gates are needed for each data signal coming from the FPGA. Therefore, if one wants to use a reversible circuit as an integrated part of an irreversible restoring logic circuit, this solution can easily be implemented. It will also reduce the number of contacts in the packaging. Even the classical enable $\overline{\text{oe}}$ and selection inputs s may be directly used. One only needs to provide the reversible chip with dual signals built at the conventional circuit side using, for example, two inverters to obtain the complementary signal. The two signals s and $\overline{\text{oe}}$ can be either constant values or adiabatic signals.

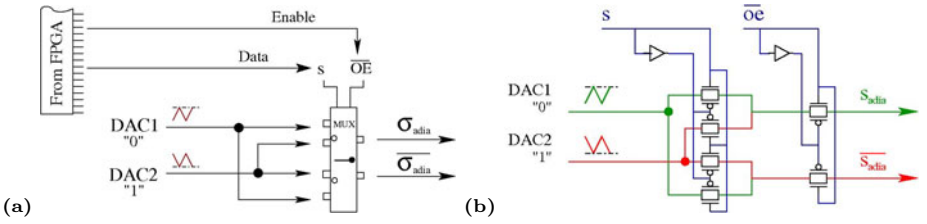

Fig. 8. (a) Demultiplexer used as signal converter. The adiabatic signals corresponding to the desired logic value defined by s, are routed from the DACs to the output, by the multiplexer. **(b)** Corresponding schematic using pass-transistor gates.

3.2 Bringing the Results of the Calculation to the FPGA

Remodeling the result represented by the adiabatic signal to a digital square-wave signal for the FPGA is less trivial. It corresponds to implement a threshold detector, which can be done using an operational amplifier. This solution is expensive, both in terms of energy consumption and surface area. In effect, if one of the dual-signal can easily be thrown away or the energy reused by implementing energy-recovering circuits, it will necessitate a buffer for impedance adaption cascaded to an operational amplifier for each signal to be send to the FPGA. This may reduce the interest for implementing reversible circuits, as the surface area or the consumption of this interface becomes bigger than the reversible circuit itself.

Instead, we can interface the adiabatic signal directly with the FPGA, as described in Sect. 2.2. This, of course, raises the question of what to do with the negated value of the dual-line adiabatic signal. To throw it away, as done in the current implementation, would dissipate energy – so perhaps this energy could be harvested with energy-recovering circuits.

3.3 The Problem of Reversibility

Controlling execution direction of the computation is essential in order to exploit the reversibility of the reversible chip, and by this way, choose the function to be performed; in this case an addition or a subtraction. A technical solution for controlling execution direction is again to use $1 \rightarrow 2$ multiplexers to route each line of the adiabatic signals to either an input or an output of the reversible chip.

The schematic in Fig. 9 presents this solution. As previously, the input *enable* allows to isolate the outputs from the inputs of the multiplexer. The *selection* input s is used as a selection bit for the execution direction. This control comes directly from the FPGA, as it is a digital signal.

Each dual-line wire of the input information is routed either to an input or to an output of the reversible chip, depending on the value of s. Our solution is simply to connect the multiplexer's first output to the reversible chip's inputs and the multiplexer's second output to the reversible chip's outputs. The inverse is done to read the computed information: the multiplexer's second output is

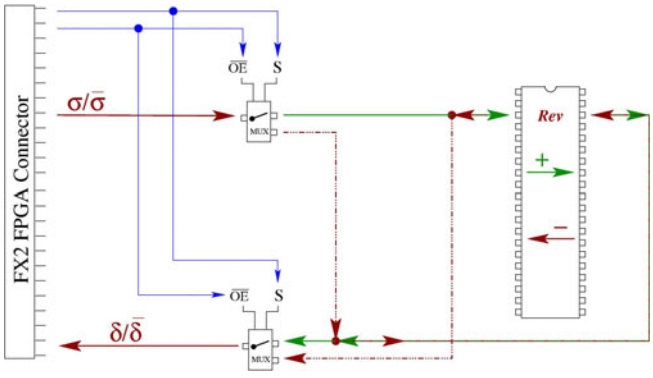

Fig. 9. Schematic showing how to use the circuit in reverse calculation. Inputs and outputs are swapped.

connected to the reversible chip's inputs and the multiplexer's first output to the reversible chip's outputs. In this configuration, the unused multiplexer outputs act as high-impedance nodes such that no short circuit occurs between the inputs and outputs of the reversible chip.

3.4 Clocking the Signals

In classical switching technology an external clock is often used. This clock signal is in the reversible DLPT technology somewhat "embedded" in the triangular shape of the input signals. The full computation performed in the reversible chip is "synchronous", but the signals propagate in an asynchronous manner from one gate to the next. In other words, the triangle waves propagate through the different gates and should arrive at the same time at the next stage of computation, without the intervention of an external clock. In practice this can not be completely guaranteed.

The clocking interfacing from the classical CMOS computing stage to the reversible computing stage, may be done by generating the dual-line triangular adiabatic signal from the classical clock, by, first, using an integrator circuit to generate one triangular pulse V_{in}, followed by a full-wave rectifier to generate the adiabatic logic "1" pulses. A simple solution to transform triangular pulses into dual-line adiabatic ones is presented in Fig. 10 [8]. These pulses are then redirected afterwards, following the method described in Section 3.1.

To synchronize the adiabatic signals coming out of the reversible stage to the classical stage is more complicated. As a solution, the amplitude of one chosen output is detected using a threshold detector and stored until the data is used. This can be done externally using flip-flops in the classical stage, after converting the adiabatic signal into classical square-wave signal, as explained in Section 3.2.

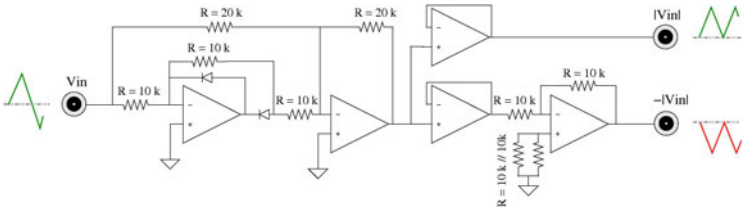

Fig. 10. Schematic of the simple circuit experimentaly used to generate the dual-line adiabatic triangular pulses from one simple triangular input

4 Conclusion

We proposed in this paper, several possible answers to the most important questions regarding the interfacing of *reversible adiabatic dual-line pass-transistor CMOS* chips with *classical restoring* circuits:

- the conversion of the pulse shape,
- the transmission of the computed information back to the FPGA,
- the reversibility of the computation,
- and the synchronization of the signals at the output interface.

For each one of these questions, at least one simple solution has been detailed and experimented, proving that embedding reversible adiabatic chips into conventional restoring circuits may be a viable yet simple solution for both, energy saving and reversibility of computation.

Acknowledgement. The authors thank the Danish Council for Strategic Research for support of this work in the framework of the MicroPower research project and the Technical University of Łódź for collaboration through *Erasmus Exchanges*.

References

1. Landauer, R.: Irreversibility and heat generation in the computing process. IBM Journal of Research and Development 5(3), 183–191 (1961)
2. Von Neumann, J.: Theory of self-reproducing automata, p. 66ss. University of Illinois Press, Urbana (1966)
3. Bennett, C.H.: Logical reversibility of computation. IBM Journal of Research and Development 17(6), 525–532 (1973)
4. Fredkin, E., Toffoli, T.: Conservative logic. International Journal of Theoretical Physics 21(3-4), 219–253 (1982)
5. Feynman, R.P.: Quantum mechanical computer. Optics News 11, 11–20 (1985)
6. Desoete, B., De Vos, A., Sibiński, M., Widerski, T.: Feynman's reversible logic gates, implemented in silicon. In: Proceedings of the 6th International Conference on MIXed DESign of Integrated Circuits and Systems (MIXDES), Kraków, pp. 497–502 (June 1999)

7. Van Rentergem, Y., De Vos, A.: Optimal design of a reversible full adder. International Journal of Unconventional Computing 4(1), 339–355 (2005)
8. Burignat, S., De Vos, A.: Test of a majority-based reversible (quantum) 4 bits ripple-carry adder in adiabatic calculation. In: Proceedings of the 18th International Conference on MIXed DESign of Integrated Circuits and Systems (MIXDES), Gliwice, Poland, pp. 368–373 (2011)
9. Amirante, E., Fischer, J., Lang, M., Bargagli-Stoffi, A., Berthold, J., Heer, C., Schmitt-Landsiedel, D.: An ultra low-power adiabatic adder embedded in a standard 0.13 μm CMOS environment. In: Proceedings of the 29th European Solid-State Circuits Conference (ESSCIRC 2003), pp. 599–602 (2003)
10. De Vos, A., Burignat, S., Thomsen, M.K.: Reversible implementation of a discrete integer linear transformation. Journal of Multiple-Valued Logic and Soft Computing 18(5), 25–35 (2011)
11. Cuccaro, S., Draper, T., Moulton, D., Kutin, S.: A new quantum ripple-carry addition circuit. In: Proceedings of the 8th Workshop on Quantum Information Processing, Cambridge (June 2005); arXiv:quant-ph/0410184v, 19 pages (2004)
12. Thomsen, M.K., Glück, R., Axelsen, H.B.: Reversible Arithmetic Logic Unit for quantum arithmetic. Journal of Physics A: Mathematical and Theoretical 43, 382002ss (2010)
13. Xilinx: Spartan-3E FPGA Starter Kit Board User Guide, UG230 (v1.1), 166 pages, June 20 (2008)
14. Younis, S.G., Knight, J.T.F.: Asymptotically zero energy computing split-level charge recovery logic. In: Proceedings of the International Workshop in Low Power Design, pp.177–182 (1994)
15. Vieri, C.J.: Reversible computer engineering and architecture. PhD. Thesis, MIT 165 pages (1999)