Available online at http://ee.istanbul.edu.tr/eng/jeee

IU-JEEE Vol. Year and Pages

1 blank line (12-point font with single spacing)

A NOVEL REVERSIBLE FAULT TOLERANT MICROPROCESSOR DESIGN IN AMS 0.35UM PROCESS

1 blank line (12-point font with single spacing)

M. Hüsrev CILASUN[1], Mustafa ALTUN [2]

2 blank line (12-point font with single spacing)

[1]Aselsan A.Ş., Gölbaşı, Ankara, Turkey
mhcilasun@aselsan.com.tr
[2]Istanbul Technical University, Maslak, Istanbul, Turkey
altunmus@itu.edu.tr

1 blank line (12-point font with single spacing)

Abstract: *In this study, reversible circuits are revisited to achieve extreme soft-defect awareness in classical CMOS circuits. Defect models in the literature are reviewed and defect scattering is analyzed. A reversible 8-bit full adder is designed in 12-bit block code domain. As a proof of concept, a pair of reversible ALUs are embedded into a microprocessor with block-code encoded data-path. The design is simulated in ams 0.35u process and a layout is obtained for tapeout.*

Keywords: *Reversible Computing, Fault Tolerance, Microprocessor.*

1 blank line using 12-point font with single spacing

1. Introduction

1 blank line (10-point font with single spacing)

Fault tolerance is an important concept for critical circuits operating in harsh conditions with high reliability demands. When a bit error is occurred in runtime, it is very hard to detect and fix the abnormal circuit behavior. In most cases, soft errors happen instantaneously and it is unlikely that a permanent fault pattern is observed. Soft error concept has been known since early 1970s and it might be caused by various phenomena.

In very early years of satellites, several errors in circuits are observed [1]. However, these errors were not related to charge accumulation in capacitors due to solar winds. The errors were found to be as a consequence of high energy particles in deep space. Although the initial attitude is that the terrestrial circuits are safe against these heavy ions which cannot survive in the world’s atmosphere, further research has shown that these ions can trigger a reaction chain which eventually produces failures in sea-level electronic circuits [2]. Although the error rate of one error in several years might sound trivial, such a failure might be extremely critical for military or space applications. Alongside these, radioactive impurities in either packaging or doping material can also result in soft errors [3].

Alpha particle emission from the impurities in the packaging material, high-energy protons and neutrons triggered by cosmic rays, thermal neutrons, random background noise, and signal integrity (SI) problems might cause soft errors. Soft errors, such as SEU (Single Event Upset) and SET (Single Event Transient) and their correction has critical significance in terms of space environment considerations. In this sort of applications, the circuit is designed using DMR (Dual modular redundancy) and TMR (Triple modular redundancy) that is widely utilized as an irreversible design methodology, yet these approaches, combined with the inherent fault tolerance of traditional CMOS logic design, makes it harder to track and detect error patterns throughout the signal. In the literature, many circuit synthesis algorithms and test methodologies are developed and several realization techniques are suggested. In this paper, these areas will be combined, by taking CMOS reversible circuit realization as a base approach, a new circuit synthesis method is developed. Using this method, an 8-bit microprocessor is designed by considering previously suggested test methods. The design will include a fault tolerance model by taking advantage of block code mapping which will lead to the detection of multiple errors. Thus, the processor will be available for on-line testing. After optimizing the design and obtaining the realization of the circuit, testing process will begin.

To keep the paper self-containing, several preliminary information on reversible circuits will be presented. A reversible circuit is traditionally defined as a bijective mapping between two identical *n*-dimensional Boolean spaces. As their name suggest, the very basic distinctive property of reversible circuits is the availability of a backward mapping for any possible input combination in contrast with conventional combinatorial logic circuits. As a property inherited from quantum circuit design, reversible circuits are mostly considered as combinatorial cascades of reversible gates. Reversible circuits and gates can be represented in various ways. The most common form is the reversible circuit diagram which shows the cascades of reversible gates that are applied in corresponding lines. However, equivalent gate and circuit functionality can be described as a permutation matrix, a decision diagram as well as a truth table. If the gate diagram is used, an *n*-line logic vector can be propagated from the input to the output by

Received on:
Accepted on:

applying the function of each corresponding gate. If the permutation matrix representation is considered, the overall circuit functionality can be obtained as follows: if the gates are in series, a cross-product of gate matrices is considered. If the gates are in parallel, a tensor (Kronecker) product of gate matrices is computed.

The very first universal reversible gate is a CCNOT or a Toffoli gate which is proposed by Tomasso Toffoli in 1980 [4]. Toffoli gate has three inputs and three outputs. It basically flips the third input if the first two inputs are both one. By modifying the inputs to make its truth table corresponding to AND and NOT gates, one can easily show that the Toffoli gate is universal. Toffoli gates can be generalized by increasing the number of control lines (which are the black dots in the first two lines) as well as the number of targets. In [5], multiple-target Toffoli gates, namely mEXOR gates are introduced for this purpose. Toffoli gates are at the main focus of reversible circuit research since they are convenient for synthesis and easy to implement.

Scientific interest on reversible circuit synthesis dramatically increases with the introduction of quantum computing [6,7,8] since the traditional circuits are believed to be trapped by Moore's limit [9]. Since the reversibility is a must for quantum circuits due to physical obligations, numerous synthesis methods have been offered [5,10,11,12]. In synthesis, the essential concern is minimizing the number of gates, number of lines, and keeping the synthesis time as small as possible. However, the circuit is often described as truth tables whose lengths are proportional to the exponent of the number of lines, which makes the synthesis runtime infeasible for considerably high number of lines [13]. Also, reversible circuits are believed to put a lower - theoretically zero - power limit to logic circuits [14,15,16]. Reversible CMOS circuits are previously realized in pass-transistor logic with no promise on a lower bound in power consumption [17,18].

In their seminal paper in 2003, Miller *et al* proposes a novel idea for reversible circuit synthesis [10]. Their algorithm takes the truth table as an input, and processes row by row. In each row, it is guaranteed that at least one row is matched between the input and the output. Although this method is far from optimal, it is good in terms of the number of lines and avoiding garbage generation. Further transformation based methods have also been developed. For instance, in [10] there are given cyclic equivalency relations of certain cascades of Toffoli gates. A *template*, is defined as a Toffoli network whose function is an identity mapping. When cascades are replaced with the smaller template counterparts, size of the overall circuit is reduced significantly.

Although the transformation based methods are practically useful, they represent the initial data as truth tables which are exponential in size. Very long synthesis durations can be reduced [19], yet different approaches are required for a circuit-level optimality. The size of the data complicates getting a more compressed representation of the same functionality. In [11], reversible circuits are expressed in terms of binary decision diagrams and their synthesis results are better than previously proposed methods in terms of the size and computation time, which makes reversible circuit synthesis scalable especially for large functions. In [12], the decision diagram structure gets complicated, which allows a general set of Quantum circuits to be synthesized in a scalable manner. These diagrams are called QMDD, which stands for Quantum Multiple-valued Decision Diagram. Table 1 summarizes different methods and their time complexities. In designing our microprocessor, we have benefited from these techniques.

Table 1. Worst-case complexities the four possible ways to represent reversible circuits

Method	Complexity
Reversible Circuit Diagram	$O(mn)$
Truth Table	$O(2^n)$
Permutation Matrix	$O(2^{n^2})$
QMDD	$O(n)$

This paper is organized as follows. In Section 2, we give introductory information and make analysis on defects in reversible circuits. In Section 3, we present our microprocessor design. In Section 4, we present simulation results and elaborate on them. In Section 5, we discuss our contributions and future works.

1 blank line (10-point font with single spacing)

2. Defects and Reversible Circuits

1 blank line (10-point font with single spacing)

A *defect*, *fault* or *error* is a generic concept which describes undesired behavior of designed model due to internal or external abnormalities. In the content of electrical circuits, a defect is frequently related to unexpected voltage, current, charge or flux characteristics. Fault analysis is a well-studied topic for traditional logic circuits. The main classification of defects is based on the transience of the behavior. If the fault happens only for once and unlikely to reproduce, it is called a *soft error*. Similarly, if the fault causes a permanent change in circuit behavior, the effect is a *hard defect*. One of the most studied types of soft defects is Single Event Upset (SEU) which describes a bit flip in a node of a logic circuit. For a stricter and more detailed classification of soft errors in irreversible circuits, the reader is referred to [3].

From the quantum computing perspective, the error is mostly due to local decoherence or quantum noise. Since redundancy is not allowed [20], quantum computers rely on different error correction schemes in order to operate properly. Although the irreversible circuit faults are classified both in abstract and technology dependent manner, reversible circuit defects are only conceptually classified due to uncertainty or immaturity of reversible implementation schemes.

In a reversible circuit, bit flipping simply refers to the logical inversion of the value of a certain node, among a large variety of fault models. Since bit flipping is generic and can be generalized for modelling other defect models, it will be considered as the base model throughout this paper.

Since witnessing a soft defect in conventional circuits is extremely rare event, conventionally errors are modeled as a single fault in overall circuit. However, especially from the reversible circuit perspective, a single error is not necessarily what is observed at the output. Before proposing a way to detect or correct errors that can happen in reversible circuits, it is essential to investigate the quantitative behavior of a single error. When we inject random single errors to four different circuits, the number of errors at the output varies as the histogram in Figure 1 suggests.

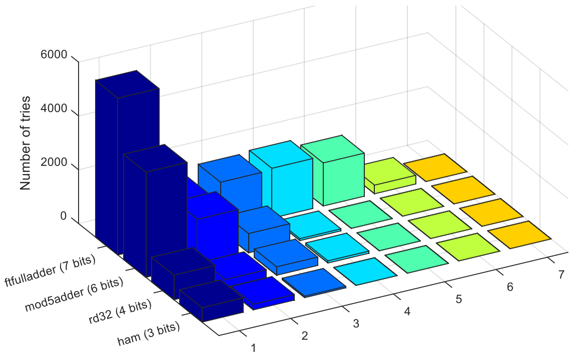

Figure 1 Unnormalized fault scattering histograms of four testbench circuits. Circuits ham, rd32, and mod5adder are taken from [21].

Since the logic applied in reversible circuits is inherently traceable, reversible circuits have a great potential to detect erroneous patterns that occur during circuit operation. In one of the earliest work on reversible fault awareness, Parhami [22] suggests parity preserving reversible gates for detecting erroneous circuit operation. In [22], it is proposed that gates such as Fredkin Gate (FG) and Feynman Double Gate (FRG) can be used for fault awareness since they have parity preserving property which means parity of inputs and outputs are equal to each other for any possible input applied to the circuit. Since each parity preserving gate is considered as a black box, the overall circuit consists of parity preserving gates will also be parity preserving. If there is a single bit flip at any node of the circuit, the corruption in the intermediate parity will propagate to the output, which will eventually cause a mismatch between input and output parities. However, there are several issues on so-called "fault-tolerance" proposed in [22]. Similar to the several papers [23] in the literature, "fault-tolerance" term is loosely defined since it actually implies an "awareness" of the fault pattern, such as described in [24]. It would be a more accurate to classify parity-preserving gates as "defect-aware" or "error-detecting" entities. Alongside this, since the parity-preserving gates are considered as black boxes, it is assumed that only one bit can be erroneous, thus no internal gate fault can result in multiple faulty lines at the output of a parity preserving gate. However, this technology-independent abstract consideration ignores the fact that the newly defined parity-preserving gates are too complex to be just a "black box". Recent parity-preserving gates can be simulated using several Toffoli gates. If an error occurs at the middle of a Toffoli cascade, there is no guarantee that it will not scatter into multiple bit errors at the output of the gate. Therefore, the initial single error assumption collapses.

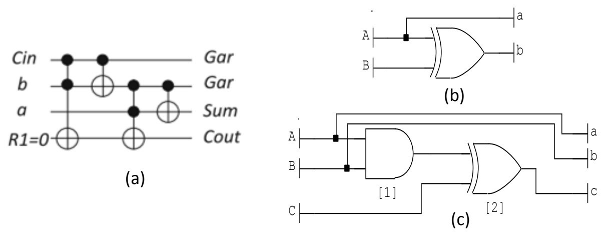

Figure 2 (a) A reversible full adder, adapted from [21]. (b) CMOS implementation of CNOT (Feynman) gate. (c) CMOS implementation of CCNOT(Toffoli) gate.

Consider the reversible circuit in Figure 2(a) and assume that it is implemented using CNOT and CCNOT gates presented in (b) and (c), respectively. If the error is at the inputs of XOR gates, the error will be seen at the output. If the error is at one of the inputs of the AND gate, AND gate might tolerate the error, yet still the fault is propagated to the output. Therefore, reversible CMOS ensures perfect fault awareness while conventional CMOS fails to accomplish this.

3. Microprocessor Design

In order to prove the concept which has been discussed in the previous chapter, a microprocessor is a comprehensive challenge. Since a low-budget fabrication is also planned, the microprocessor should be kept as small as possible. The resulting chip will be sent to *Europractice* for Multiple Project Wafer (MPW) runs which enables relatively cheap prototyping. The designed circuit will utilize a single memory for both instructions and data, therefore it can be considered to have an unpipelined Von Neumann architecture. This is illustrated in Figure 3.

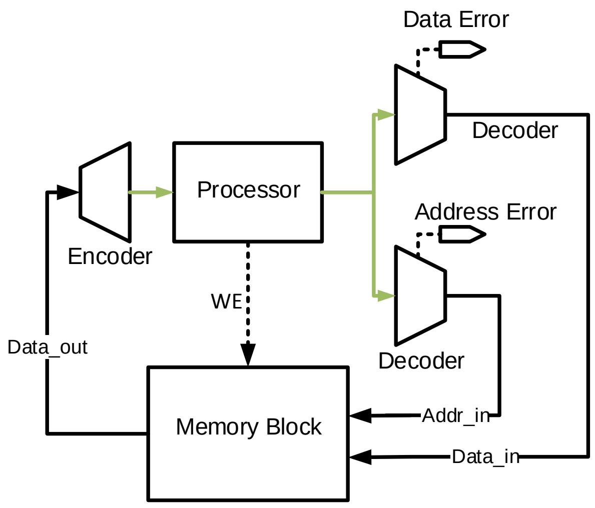

Figure 3 The top operating scheme of the CPU. Clock and reset input pins are not shown. The green path is the way where 8-bit data flow is 12-bit block-code encoded. The dashed signals are single wires which are not included in a bus.

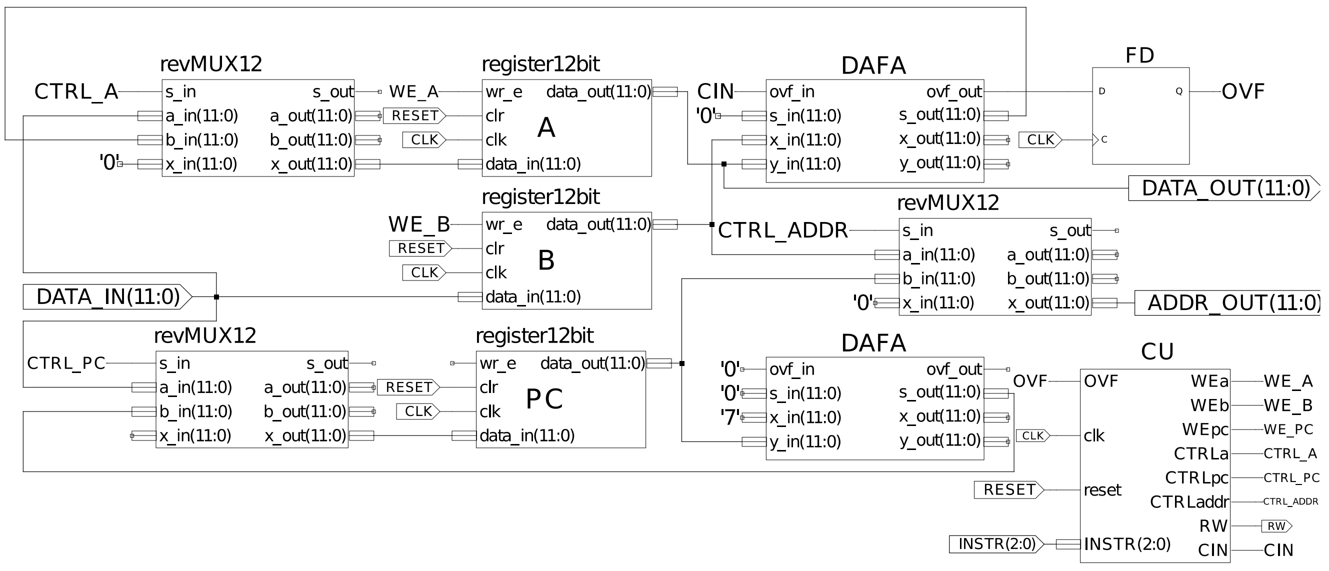

Figure 4 The functional block diagram of the microprocessor.

The instruction set is as small as possible to be operated sufficiently such that the device is classified as a RISC processor. In order to achieve fault-tolerant property, the input will be given as a 12-bit bus instead of actual 8-bit data bus in a block code encoding, as previously proposed for quantum circuits [25]. The accumulator A, register B and program counter register PC will store data and address information as a 12-block code. The ALUs will also be capable of performing reversible addition and subtraction of 12-bit block code encoded 8-bit unsigned integers. Due to the reversibility, no errors at the ALUs will disappear as discussed before. Also, the errors at registers will also be propagated to the output. The functional block diagram of the microprocessor is given in Figure 4.

In the practical operation scheme, a block code encoder and decoder will accompany the microprocessor alongside a Von Neumann styled memory. The top operation scheme is illustrated in Figure 3 where, the encoder performs block code mapping of the 8-bit input data or address onto 12-bit block code domain. When the data are being read from the microprocessor, two decoders perform maximum likelihood decoding to recover the data. Since $d = 3$, decoders will correct one error and also raise an output signal up to two errors, indicating that the output might not be reliable. The memory has a 256 bytes of storage with positive write enable input and positive asynchronous reset. The three most significant bit of the 8-bit raw output of the memory is the instruction input of the microprocessor while the encoded part is connected to the data-in pins.

The design includes three conventional 12-bit multiplexers. In future work, MUXes are planned to be replaced with reversible counterparts in order to ensure maximum defect awareness. Multiplexers are controlled with control signals CTRL_A, CTRL_PC and CTRL_ADDR by the control unit (CU).

When CTRL_A signal is low, the input of the accumulator is connected to the output of the ALU and when CTRL_A is high, input of the accumulator is received from the external data input. If CTRL_PC is low, program counter loads the 1-incremented value of itself from the previous cycle and it loads the external address if otherwise. If CTRL_ADDR is low, the output address bus forwards the content of the program counter PC. If it is low, then content of the register B is forwarded. WE_A, WE_B and WE_PC signals are the Write-Enable inputs of the A, B, and PC registers, respectively. All registers have positive clock and positive asynchronous reset inputs as well. The external read/write signal RW and ALU's input carry signal, CIN are also driven by the control unit. If the subtraction will be performed, CIN is raised, but the second input B is expected to be stored as an inverted manner since the ALU has not controllable inverter. There are two reversible DAFA modules where the first DAFA is the arithmetic logic unit connected to A and B registers. The second DAFA module is connected to the PC register and a constant-1, thus it behaves as a program incrementor.

Table 2. Instruction set of the microprocessor.

Instruction	**Operation**	**Cycles**
ADD [000x xxxx]	A <= A+B PC <= PC+1	2
SUBTR [001x xxxx]	A<=A-INV(B) PC <= PC+1	2
LDA [010x xxxx; DATA_IN]	A<=DATA_IN PC <= PC+2	2
LDB [011x xxxx; DATA_IN]	B<=DATA_IN PC <= PC+2	2
STR [100x xxxx; ADDR_IN]:	B<=ADDR_IN PC <= PC+2 Enable W/R	3
JMP [101x xxxx; ADDR_IN]	PC<=ADDR_IN	2
JMPOVF [110x xxxx; ADDR_IN]	PC<=ADDR_IN Overflow==1	2
HALT [111x xxxx; ADDR_IN]	Halts execution.	1

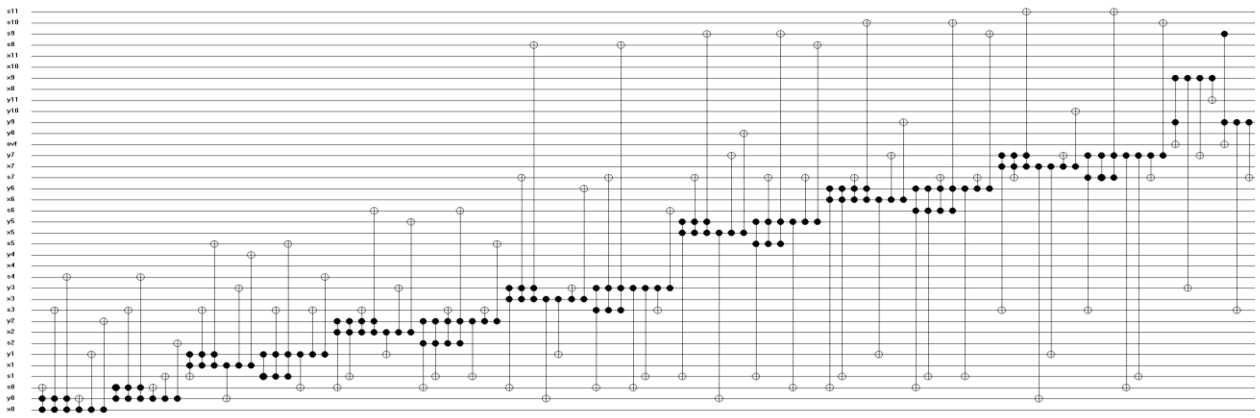

Figure 5 Reversible block code mapped fault tolerant 8-bit full adder based on [26].

The final reversible circuit of the processor composed of Toffoli gates is shown in Figure 5.

The microprocessor has eight instructions as given in Table 2. Although the instruction set is quite limited, it is sufficient to be a proof of concept. The control unit is described in Verilog hardware description language. Four internal 1-bit state registers are utilized to implement total 10 states. Alongside the positive triggered asynchronous reset and clock signals, control unit also receives 3-bit instruction bus and 1-bit output signal from the overflow flag as its inputs. Figure 6 represents the finite state diagram of the proposed control unit.

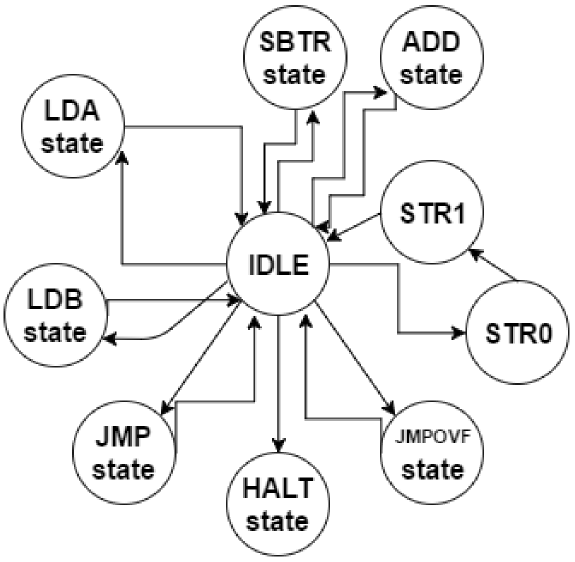

Figure 6 Finite state diagram of the control unit. Triggering control signals are omitted.

In order to directly perform block code mapping to obtain a fault tolerant circuit, one would require 22 bits to encode 17 bits of logic. However, this will cause decoder circuit to be large enough to decode 4 megabytes of decoding information, which will be infeasible to realize. To overcome this problem, 8-bit data buses x, y, and s are transformed to the 12-bit block code domain separately and we implemented our mapping algorithm to three data buses. The resulting circuit has 37 lines and 100 gates with 288 quantum cost while the original circuits had 32 gates with 64 cost. In the implementation, the gates with identical control lines are merged to reduce actual CMOS cost. The design is also verified in Xilinx ISE using a Verilog test fixture code.

The processor is synthesized in Cadence Encounter RTL Compiler using ams 0.35µ C35B4 process digital core library. From the RTL schematic, it can be observed that the total power consumption of the microprocessor is $1.68\ mW$. Examining the timing report, the maximum fanout is 5, maximum load capacitance is $74.8\ fF$, the maximum slew is $2248\ ps$, and the maximum delay is $1039\ ps$. The total area is $0.046\ mm^2$.

4. Simulation and Verification

After the analysis of the synthesized we simulated the microprocessor using the test code in Table 3. A Verilog test fixture code which initializes and maintains the reset and clock signals is executed in ISim simulator which is embedded inside Xilinx ISE. Contents of the memory locations during the execution of the program whose waveform pattern is given in Figure 7 are provided as follows:

Table 3 Test code for the microprocessor.

Memory Location	Opcode	Memory Content
R[0]	LDA	0100 0000
R[4]	#10	0000 1010
R[5]	LDB	0110 0000
R[6]	#5	0000 0101
R[10]	ADD	0000 0000
R[11]	STR	1000 0000
R[12]	@8	0000 1000
R[21]	HALT	11100000
R[13]	***	0000 0000
R[13]$^+$	#15	0000 1111

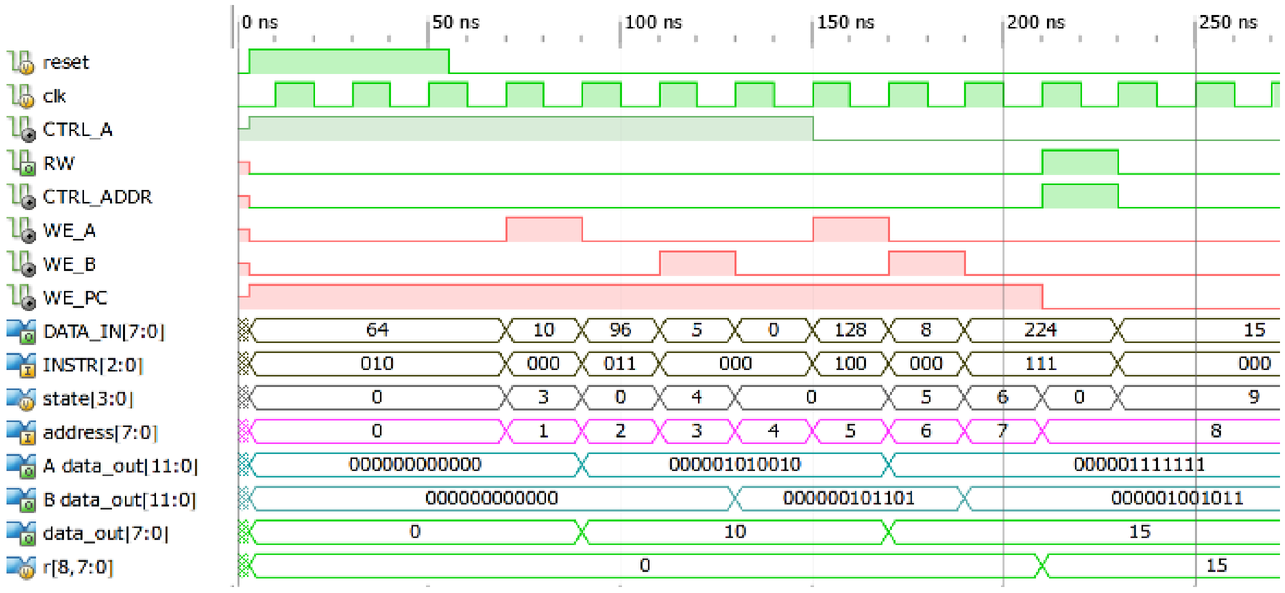

Figure 7 RTL simulation results for a test program.

After the program in Table 3 is executed, the resulting value 15 is written to the memory location R[13] while its updated value is denoted as R[13]$^+$. Memory contents can be followed at DATA_IN bus of Figure 7 in decimal radix. After the execution is done, the microprocessor performs a transition to the HALT state, thus no further opcode will be executed.

Upon the verification of the design, technology dependent post-synthesis Verilog code is modified to avoid logic trimming of our block code mapped fault tolerant reversible full adder. Alongside the design constraints (SDC) file, the resulting Verilog design is transferred to the Cadence Encounter Place and Route environment. The design was floorplanned and power routing by adding cell rings was performed. After adding three power stripes, end capacitors are added in order to achieve decoupling, i.e. electrically separating the substrate and wells by limiting the resistance in between them. After this step, cell blocks and IO pins are placed. No special handling for clock tree is performed since the expected clock delay is not critical considering the planned operating frequency. Afterwards, unused spaces are being filled with dummy metal. Finally, the design is automatically routed, generating 0 violations.

After the layout is obtained, post-layout report is examined. The resulting chip has 42 pins which consists of 12 DATA_IN, 12 DATA_OUT, 12 ADDRESS_OUT, 1 CLK, 1 RESET, 3 INSTR signals. Total number of utilized standard cells is 628, including left and right hand side end capacitors and fill shapes. The report also extracted the floating outputs (s and y) of ALU, which are generated as a sequence of reversibility. There are four routing layers MET1, MET2, MET3, and MET4.

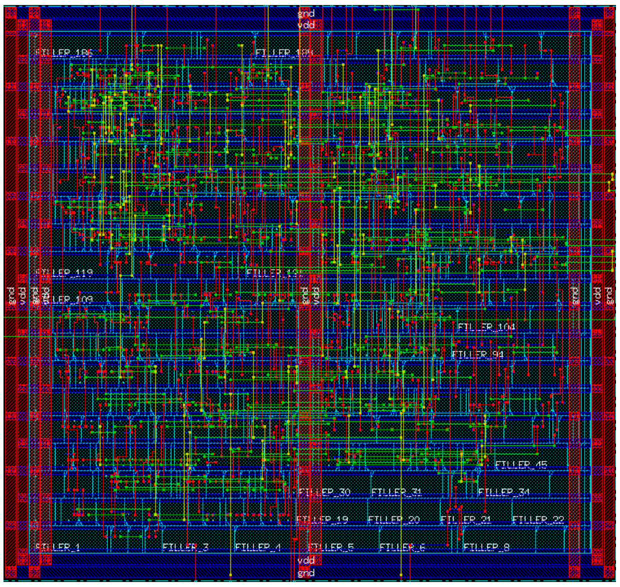

Figure 8 Encounter post-place-and-route view of the microprocessor.

The post-place-and-route and the layout pictures of the proposed microprocessor are given in Figure 8 and Figure 9, respectively. The whole chip occupies 0.08 mm^2 area. In the standard cell mapping, 19 rows are used with a gate density of 59.99% excluding physical cells. The core utilization of the whole chip is 83.22% in terms of standard cells, IO, and macro blocks. The total wire length is 23981.625 μm with an average wire length of 43.2881 μm per net. The detailed area numbers are given in Table 4.

Table 4. Detailed area report of the microprocessor.

Total area of Standard cells:	67085.200 μm^2
Total area of Standard cells (Subtracting Physical Cells):	40276.600 μm^2
Total area of Core:	67128.425 μm^2
Total area of Chip:	**80614.000 μm^2**

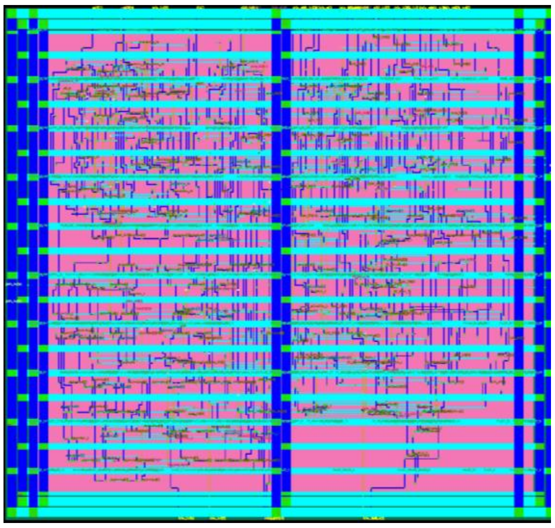
Figure 9 Negative GDSII layout of the microprocessor.

5. Conclusions

1 blank line (10-point font with single spacing)

In this study, a fault-tolerant and defect-aware reversible RISC microprocessor has been designed as the proof of the concept where we show that reversible computing can be utilized to achieve perfect defect awareness. The design will be sent to fabrication to conduct further tests on the dual in-line packed tapeout. Prior to the submission of the design for tapeout, the ALU will be verified for a complete set of possible inputs. Multiplexers and control logic will be made reversible and/or defect-aware. The design will be carried out further design-rule checks. Post-layout simulation will be conducted including the delays resulting from parasitic capacitances and resistances.

6. References

1 blank line (10-point font with single spacing)

[1] Binder D., Smith, E.C. and Holman, A.B., 1975. “Satellite anomalies from galactic cosmic rays”, IEEE Trans. Nucl. Sci., vol. NS-22, no. 6, pp. 2675–2680.

[2] May, T.C. and Woods, M.H., 1978. “A new physical mechanism for soft errors in dynamic memories”, in Proc. Int’l Rel. Phys. Symp. (IRPS), pp. 33–40.

[3] Nicolaidis, M. ed., 2010. Soft errors in modern electronic systems (Vol. 41). Springer Science & Business Media.

[4] Toffoli, T., 1980. J. W. de Bakker and J. van Leeuwen, ed. "Reversible computing". Automata, Languages and Programming, Seventh Colloquium. Springer Verlag, Noordwijkerhout, Netherland.

[5] Maslov, D., Dueck, G. W., and Miller, D.M., “Toffoli Network Synthesis with Templates. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Vol. 24, No. 6, June 2005

[6] Brown, J., 2000, The Quest for the Quantum Computer, Touchstone, New York.

[7] Benioff, P., 1980. The computer as a physical system: A microscopic quantum mechanical Hamiltonian model of computers as represented by Turing machines. Journal of Statistical Physics, 22(5), pp.563-591

[8] Beckman, D., Chari, A.N., Devabhaktuni, S., and Preskill, J., 1996. Efficient networks for quantum factoring. Physical Review A, 54(2), p.1034.

[9] Moore, G.E., 1965. Cramming more components onto integrated circuits, Electronics Magazine. p. 4.

[10] Miller, D.M., Maslov, D., and Dueck, G.W., 2003, June. A transformation based algorithm for reversible logic synthesis. In Proceedings of the 40th annual Design Automation Conference (pp. 318-323). ACM

[11] Wille, R. and Drechsler, R., 2009, July. BDD-based synthesis of reversible logic for large functions. In Proceedings of the 46th Annual Design Automation Conference (pp. 270-275). ACM.

[12] Miller, D.M. and Thornton, M.A., 2006, May. QMDD: A decision diagram structure for reversible and quantum circuits. In Multiple-Valued Logic, 2006. ISMVL 2006. 36th International Symposium on (pp. 30-30). IEEE.

[13] Soeken, M., Wille, R., Hilken, C., Przigoda, N., and Drechsler, R., 2012, January. Synthesis of reversible circuits with minimal lines for large functions. In Design Automation Conference (ASP-DAC), 2012 17th Asia and South Pacific (pp. 85-92). IEEE.

[14] Bennett, C.H., 2003. Notes on Landauer's principle, reversible computation, and Maxwell's Demon. Studies In History and Philosophy of Science Part B: Studies In History and Philosophy of Modern Physics, 34(3), pp.501-510.

[15] Landauer, R., 1961. Irreversibility and heat generation in the computing process. IBM journal of research and development, 5(3), pp.183-191.

[16] Willingham, D.J. and Kale, I., 2008. Using positive feedback adiabatic logic to implement reversible Toffoli gates.

[17] Thomsen, M.K., 2012. Design of reversible logic circuits using standard cells: Standard cells and functional programming. Department of Computer Science, University of Copenhagen.

[18] Burignat, S., Thomsen, M.K., Klimczak, M., Olczak, M., and De Vos, A., 2011. Interfacing reversible pass-transistor CMOS chips with conventional restoring CMOS circuits. InReversible Computation (pp. 112-122). Springer Berlin Heidelberg.

[19] Susam, O. and Altun, M., 2016. Fast Synthesis of Reversible Circuits using a Sorting Algorithm and Optimization. Journal of Multiple-Valued Logic and Soft Computing, accepted for publication.

[20] Wootters, W., Zurek, W., 1982. "A Single Quantum Cannot be Cloned". Nature 299: 802–803.

[21] Wille, R., Große, D., Teuber, L., Dueck, G.W., and Drechsler, R., 2008, May. RevLib: An online resource for reversible functions and reversible circuits. In Multiple Valued Logic, 2008. ISMVL 2008. 38th International Symposium on (pp. 220-225). IEEE.

[22] Parhami, B., 2006, October. Fault-tolerant reversible circuits. In Signals, Systems and Computers, 2006. ACSSC'06. Fortieth Asilomar Conference on (pp. 1726-1729). IEEE.

[23] Boykin, P.O. and Roychowdhury, V.P., 2005, June. Reversible fault-tolerant logic. In Dependable Systems and Networks, 2005. DSN 2005. Proceedings. International Conference on (pp. 444-453). IEEE.

[24] Przigoda, N., Dueck, G., Wille, R. and Drechsler, R., 2016. Fault Detection in Parity Preserving Reversible Circuits.

[25] DiVincenzo, D.P. and Shor, P.W., 1996. Fault-tolerant error correction with efficient quantum codes. Physical review letters, 77(15), p.3260.

[26] Van Rentergem, Y. and De Vos, A., 2005. Optimal design of a reversible full adder. International Journal of Unconventional Computing, 1(4), p.339.

1 blank line (10-point font with single spacing)

Note:

1 blank line (10-point font with single spacing)

Mustafa Altun received his BSc and MSc degrees in electronics engineering at Istanbul Technical University in 2004 and 2007, respectively. He received his PhD degree in electrical engineering with a PhD minor in mathematics at the University of Minnesota in 2012. Since 2013, he has served as an assistant professor of electrical engineering at Istanbul Technical University. Dr. Altun runs the Emerging Circuits and Computation (ECC) Group in the same university. Dr. Altun has been served as a principal investigator/researcher of various projects including EU H2020 RISE, National Science Foundation of USA (NSF), TUBITAK Career, and TUBITAK University-Industry Collaboration projects. He is an author of more than 30 peer reviewed papers and a book chapter, and the recipient of the TUBITAK Success, TUBITAK Career, and Werner von Siemens Excellence awards.

M. Hüsrev Cılasun is currently an engineer at Aselsan A.Ş. His research interests include reversible and quantum circuits, FPGA/ASIC design, digital signal and image processing, and machine learning. He is author of several conference and journal papers on EEG processing and autonomous direction estimation, number theory, and roboti