

Towards a Balanced Ternary FPGA

Paul Beckett

Electrical & Computer Engineering, RMIT University
Latrobe St., Melbourne, Australia
pbeckett@rmit.edu.au

Abstract—We propose and analyze an organization for a Field-Programmable Gate Array structure that operates using a balanced ternary logic system where the logic set $\{\pm 1, 0\}$ maps directly to equivalent voltage levels $\{\pm 1.0V, 0.0V\}$. Circuits for basic components such as a ternary buffer, flip-flop and LUT are described based on the characteristics of a commercial silicon-on-sapphire process that offers multiple simultaneous transistor thresholds. A simple example of a balanced ternary FIR filter is mapped to the FPGA and some preliminary estimates made of its performance and area.

I. INTRODUCTION

Balanced ternary, described by Knuth as “*Perhaps the prettiest number system of all*” [1], has been of interest to researchers for many years. However, the simplicity and ubiquity of binary has ensured that few ternary systems have been implemented commercially. At the same time, it is widely recognized that the short word-length (SWL) format generated by sigma-delta modulators often results in greatly simplified arithmetic processing for DSP functions, particularly in cases requiring many multiplication steps. In particular, using a ternary quantizer [2] to encode filter coefficients in a balanced ternary format may simplify the required operations by limiting the filter tap values to the set $\{-1, 0, +1\}$ such that multiplication operations reduce to simple logic or multiplexing functions.

This paper explores the idea of directly implementing a Field-Programmable Gate Array structure using a balanced ternary logic system where the logic set $\{\pm 1, 0\}$ directly maps to the equivalent voltage levels, $\{\pm 1.0V, 0.0V\}$. While it is relatively straightforward to map ternary operations onto an equivalent binary set, for example by representing a ternary digit with 2-bit 2’s complement binary or other double-rail logic systems [3], [4], this does not take full advantage of the compact ternary notation as well as increasing the number of interconnect lines between logic cells. The ultimate objective of this work is the development of an integrated platform (a system on chip; SoC) capable of directly support a range of general-purpose short-word-length DSP functions, including classical [5] and adaptive LMS filtering [6], [7], where both the stream data (video or audio) and the filter coefficients are represented in balanced ternary format.

A number of circuit designs are presented that represent basic components of the ternary FPGA, including a ternary buffer, latches, flip-flops and a small look-up table (LUT).

These are based on the characteristics of a commercially available 250nm Silicon-on-Sapphire process [8]. This process is capable of high-speed operation ($f_t \approx 60\text{GHz}$) and will support RF, analog and mixed signal as well as digital circuits in a complete SoC approach. Most importantly for this application the process offers multiple simultaneous transistor threshold voltages that will support the necessary ternary components. The caveat here is that this is a 1-poly, 3-metal process, which could potentially make it difficult to efficiently lay out a complex system such as a FPGA. However, the techniques are directly applicable more sophisticated (and more costly) multi-threshold SOI processes, such as the 45nm IBM 12SOI process with four simultaneous V_{TH} values offered via mosis.com [9].

The remainder of this paper proceeds as follows. Section II introduces the idea of balanced ternary arithmetic and looks at previous work in this domain. Then in Section III, the designs and simulation results for the proposed ternary FPGA components are presented. In Section IV we illustrate the operation of the proposed FPGA by mapping a small application to the device. Finally in Section V we conclude and identify future work.

II. BALANCED TERNARY

Balanced ternary representation is especially useful for describing “difference” signals ($>$, $=$, $<$) such as those derived from $\Sigma\Delta$ ADC systems and offers a number of computational advantages [1]:

- the full integer range can be represented without requiring a separate sign bit;
- the negative value of any number is derived simply by inverting all ternary digits (*trits*) (i.e., replacing +1 with -1 and vis versa);
- thus, subtraction is straightforward—just invert and add.
- similarly, single-trit multiplication (by $\pm 1, 0$) reduces to a controlled inversion stage—most easily implemented as a simple LUT;

In an early example of the use of a ternary notation, Rajashekhara and Chen [10] described a “carry-free” ternary adder based on a balanced digit set and built using a so-called ternary ‘T-gate’. This is essentially a transmission gate multiplexer controlled two majority logic comparators that have their two thresholds set mid-way between the three logic levels. In the CMOS process available at that time, the

only way to adjust the threshold levels was to modify the gain ratio of the p and n-type transistor $k_R = \frac{k_n}{k_p}$ (where $k_n = \mu_n C_{OX} W_n / L_n$, μ and C_{OX} are process parameters and k_p has an identical form). As the term $\mu_{n,p} C_{OX}$ is fixed by the process, the relative Width/Length (W/L) ratios had to be adjusted over a very wide range (0.1, 1.0 and 10 in that work). More recently, [11] described a low-power ternary full adder circuit using $2\mu\text{m}$ n-well CMOS in which well-biasing was used to adjust the switching thresholds to the appropriate levels. The issue of switching threshold adjustment will be discussed in more detail in the following section.

Various non-binary arithmetic circuits have been proposed that use floating gate mechanisms of the sort originally described by Shibata and Ohmi [12]. For example, in [13], Gundersen and Berg describe a voltage-mode balanced ternary adder circuit based on dynamic semi-floating gate devices that operates with a 1 GHz clock at $V_{DD} = 1\text{V}$ using a readily available 90nm CMOS process. However, all floating-gate circuits require some form of charge initialization that complicates their design, while a further disadvantage of this particular approach is the need to supply a high-speed clock signal with a frequency twice the data throughput rate.

Ternary representations have also been proposed as a means of reducing interconnection complexity and power loss [14], [15] as well as to reduce crosstalk in parallel interconnect wires [16]. Similarly, [17] uses ternary coding to reduce the number of lines required to control the analog multiplexers/demultiplexers in a proposed reconfigurable analog array.

A. Ternary Logic

As mentioned in the introduction, the work reported here is based on the characteristics of a 250nm Silicon-on-Sapphire (SoS) process. This has been chosen for a number of reasons:

- The process supports a number of transistor threshold values (three each for the p and n-type transistors) making it relatively easy to achieve the range of switching thresholds needed for the proposed ternary circuits. As a “fringe-benefit”, static (subthreshold) power can be traded for performance where required by applying the appropriate threshold ;
- although the feature size is nominally large (i.e., behind current state of the art) the insulating substrate removes the need for deep well structures and guard regions. As a result, devices can be packed more densely than in a corresponding bulk CMOS process. This is particularly relevant to the structure of transmission gates using both p and n-type pass transistors;
- for the same reason (insulating substrate), the process is intrinsically fast and low-power—the source/drain and interconnect capacitances are small, resulting in reduced intrinsic delay ($\propto CV_{DD}/I_D$) and dynamic power ($\propto FCV_{DD}^2$) as well as low RC interconnect delay;

In exactly the same manner as for binary, the operation of ternary logic relies on establishing a known switching threshold at the input of the receiving gate. The switching

TABLE I
SWITCHING THRESHOLDS (mV) WITH $k_R=1.36$; $V_{supply} = \pm 1\text{V}$.

	P	Intrinsic (mV)			High Threshold (mV)		
		Slow	Typ	Fast	Slow	Typ	Fast
N		-160	0	150	-750	-600	-450
Slow	250	-16	58	127	-288	-219	-148
Typ	100	-97	-23	46	-369	-300	-228
Fast	-53	-179	-105	-36	-451	-382	-311
Slow	853	309	383	452	36	106	177
Typ	700	226	300	369	-46	23	95
Fast	550	146	219	289	-127	-58	14

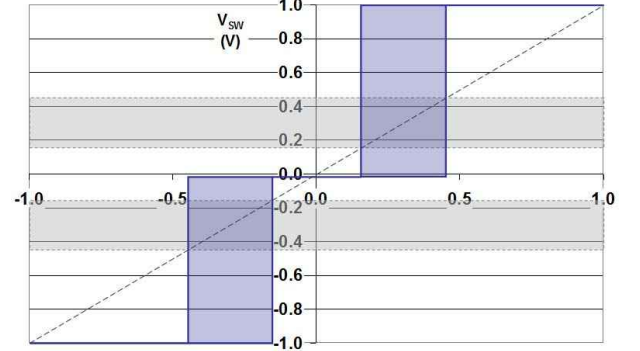


Fig. 1. Worse-case Ternary Logic Switching Thresholds and Noise Margins (data from Table 1)

threshold of a simple gate is given by [18]:

$$V_{SW} = \frac{V_{THN} + \sqrt{\frac{1}{k_R}}(V_{DD} + V_{THP})}{1 + \sqrt{\frac{1}{k_R}}} \quad (1)$$

where V_{THN} and V_{THP} are the thresholds voltages of the n and p-type transistors respectively, and k_R is the ratio defined above. The SoS process supports three transistor threshold ranges for both p and n-types: intrinsic ($\sim \pm 100\text{mV}$), regular ($\sim \pm 400\text{mV}$) and high-threshold ($\sim \pm 700\text{mV}$). Table I and Fig. 1 show the switching thresholds (V_{SW}) derived using (1) with intrinsic and high-threshold values across their typical, fast and slow corners and with k_R fixed at 1.36 so that the typical switching thresholds are symmetrical at $\pm 300\text{mV}$ or about 30% of the supply (± 1). Threshold variability results in a V_{SW} band $\sim \pm 150\text{mV}$ either side of its typical value (Fig. 1). Obviously, the worse-case noise margin ($\pm 150\text{mV}$ around 0V) can be further improved by adjusting k_R to move the switching points away from zero at the cost of an increased asymmetry between the switching regions.

Techniques for balanced ternary to binary conversion are well known. For example, the organization of the inverter pair shown in Fig. 2 is similar to the Positive and Negative Ternary Inverter stages of [10] and transforms ternary levels $\{\pm 1, 0\}$ to binary in the range $\{-1, +1\}$. Using the typical *intrinsic* and *high-threshold* values as shown in Table I, and adjusting W/L in the normal way to allow for mobility differences in the p and n-channels, V_{SW} is set at just over $\pm 300\text{mV}$ (Fig. 1). Two variations on this basic mechanism have been used to

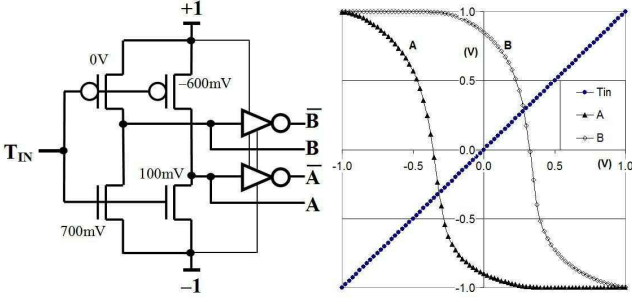


Fig. 2. Ternary-to-Binary interface and its simulated DC voltage transfer characteristics

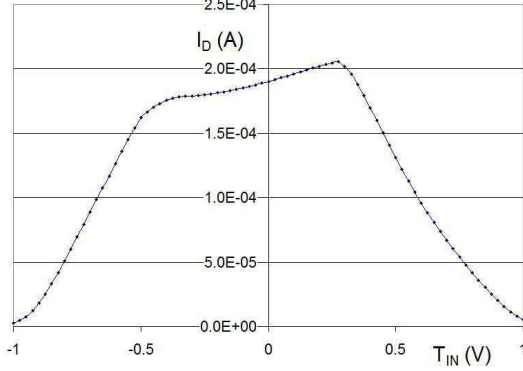


Fig. 3. DC drain current characteristic of the ternary to binary interface in Fig. 2

develop the required FPGA components as will be described in the following section.

III. TERNARY FPGA COMPONENTS

This section describes the basic components from which our proposed ternary FPGA structure will be built. At present, we envisage a conventional “island-style” organization in which basic blocks comprising a small look-up table (LUT) and flip-flops are interconnected by orthogonal signal lines linked to switch boxes. As level restoration is a fundamental requirement in any digital logic system, a ternary to ternary buffer is analyzed first, followed by the remaining FPGA components. The following simulations were carried out with Synopsys® HSpice version X.2005 using *typical* level 58 MOS models supplied by the foundry (see Table I).

A. Ternary Buffer

A significant problem with the simple Ternary interface circuit of Fig. 2 is its high standby current when detecting the center symbol, in this case at $T_{IN}=0V$ (Fig. 3). The ternary buffer circuit of Fig. 4 was developed to avoid this problem and comprises two stages. The input stage is similar to the ternary decoder of [15] and converts ternary voltages $\{-1, 0, +1\}$ to binary pairs $\{A, B\}$ as shown in the table. This stage exhibits a subthreshold leakage current in the order of 40pA at $T_{IN}=0V$ using *regular* transistors with a $|V_{TH}|$ of around 400mV. The following stage then converts these signals back into ternary as illustrated in Fig. 5. The worse-case (FO-4)

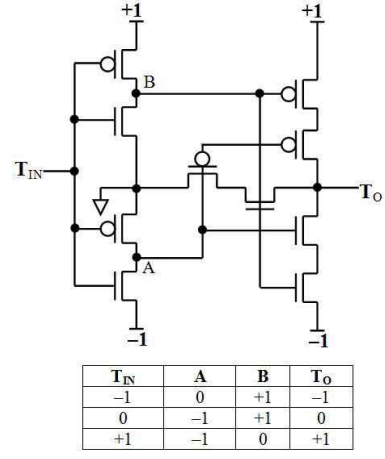


Fig. 4. Ternary buffer circuit. Input stage with $k_R \approx 1.2$ and $|V_{TH}| \approx 400mV$ for each FET gives $V_{SW} \approx \pm 0.5V$

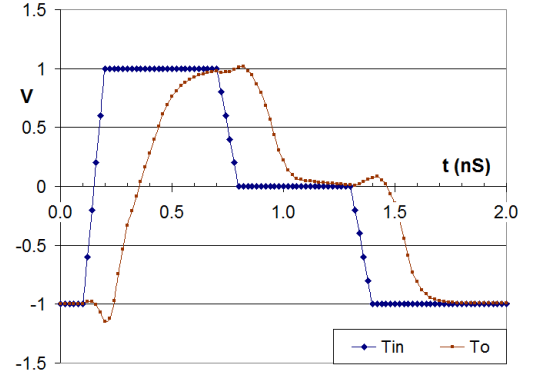


Fig. 5. Transient performance of ternary buffer with FO-4 load

delay for the buffer is approximately 200pS and occurs for the transitions $(+1 \rightarrow 0)$ and $(0 \rightarrow -1)$.

B. Ternary Latch and Flip-Flop

The ternary buffer circuit described above forms a key component in the FPGA circuits proposed here. For example, the transparent latch circuit shown in Fig. 6 simply substitutes the buffer for the non-inverting stages of a conventional latch. It is assumed that a symmetrical clock pair (*clock* and *clock*) in the range $\{-1, +1\}$ will be distributed to the logic blocks. The general performance the ternary latch is shown in Fig. 7 with a number of input and clock transitions (the *clock* signal has been omitted for clarity). In this latch, the feedback transmission gate is ‘weak’, i.e., with a L/W ratio of more than 5:1, and serves to maintain static operation of the latch.

This latch component will be used in two ways. Firstly, combining two of them in a conventional master-slave configuration, results a circuit with the timing waveform of Fig. 7 with a worse-case propagation delay of just under 300pS. Here, the propagation delay is measured from 50% of the clock transition (i.e., at 0V) to the 50% of the output waveform (i.e., 0V, +500mV or -500mV depending on the transition). In master-slave organizations of this type, the setup time is equal to the delay through the first buffer stage so is similar to the

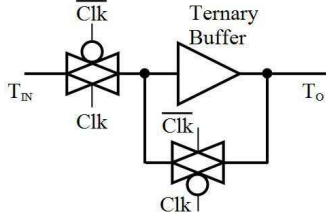


Fig. 6. Ternary Latch Circuit

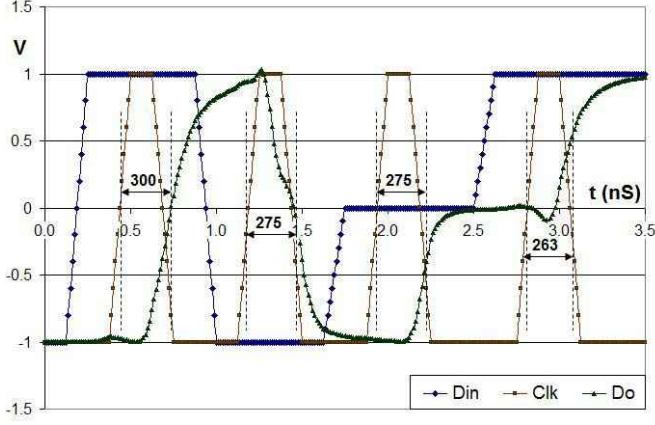


Fig. 7. Ternary master-slave flip-flop, Clock→Output delay as shown (pS, FO-4)

clock to output delay figure. This master–slave flip-flop will be used in a basic logic element (BLE) within the FPGA. Its second application as a configuration register will be described in the section III-D below.

C. Ternary Logic Element and Interconnect

The Basic Logic Element (BLE, Fig. 8) of a Field Programmable Array typically comprises a small LUT plus a flip-flop. As the organization of the BLE impacts directly on the mapping efficiency, there has been much discussion over issues such as LUT size and whether the register is always necessary or not. There is a clear tension between the size of the LUT (i.e., the number of input bits) and its area and performance. Averaged over a wide range of benchmarks the area*delay figure has been shown [19] to exhibit a shallow optimum in the region of four to six (binary) inputs. The following description is mainly based on a 3-LUT organization where the three input trits select one of 27 configuration trits. Thus a ternary 3-LUT is roughly equivalent to 4.75 bits and will directly support a medium complexity function such as a single-trit multiplier or full adder. We have also briefly analyzed the performance of a 4-LUT topology that would support trinary (three operand) operations. The area–performance tradeoffs implicit in these design decisions will be left for future work.

In the three–input multiplexer (3MUX) shown in Fig. 9a, the two series transmission gates are controlled via the ternary input pairs that have been decoded using the T2B circuit of Fig. 2. Although it was noted above that this circuit exhibits high standby (subthreshold) current in its '0' state (see Fig 3),

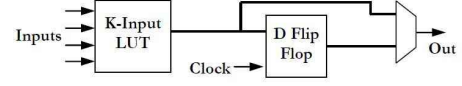
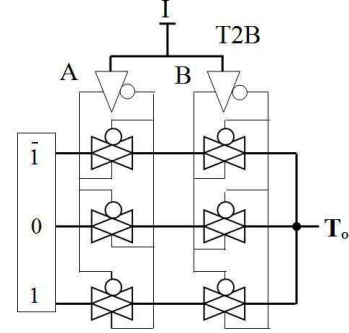
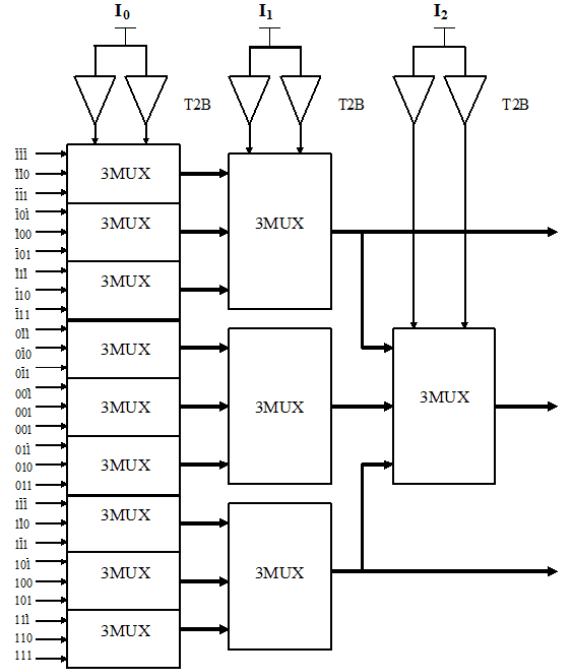


Fig. 8. Basic Logic Element Structure



(a) 3MUX: a single-trit selector based on pass-gates



(b) 3-trit LUT structure

Fig. 9. Ternary 3-LUT block diagram

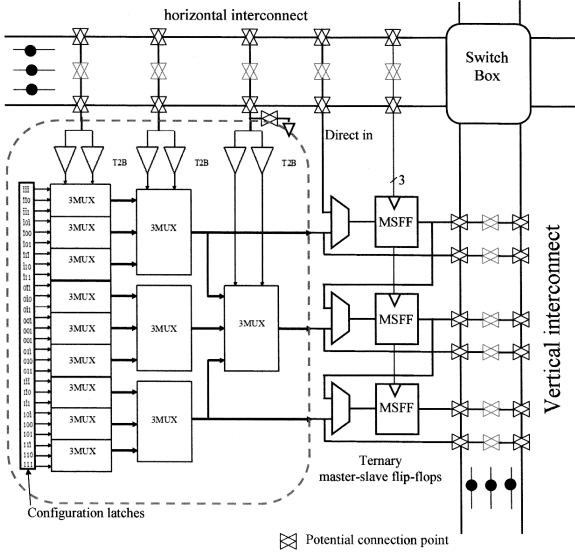
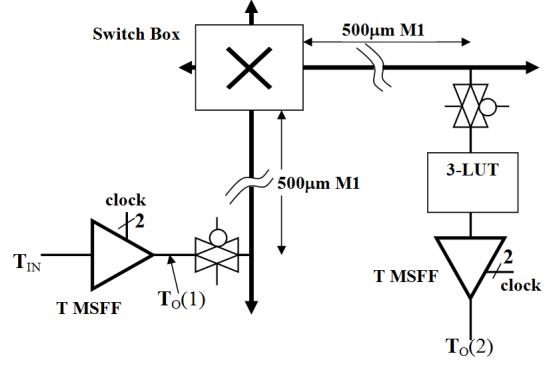


Fig. 10. Proposed Ternary Logic Block organization and local interconnect

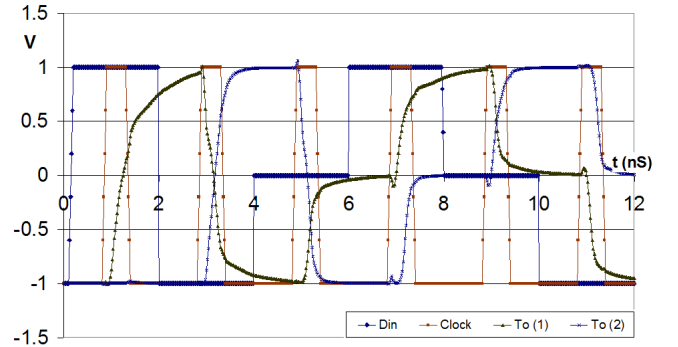
we consider this to be acceptable in this case as only a small number of the interface circuits will be used in the system (three per BLE) and logic '0' is only one of three equally probable logic states, further reducing its contribution to the average overall leakage current. Note that this situation is quite different to [15] where the center state ($V_{DD}/2$ in that case) represents the default or *inactive* condition for their ternary buss drivers. The resulting increase in static current would very likely overwhelm any savings in dynamic power/energy. The 3MUX components are used in a conventional hierarchical organization (Fig. 9b) that has been partitioned into two, 2-trit groups in a manner that supports single-trit operations such as multiplication while still allowing access to the otherwise unused flip-flops.

Fig. 10 illustrates the proposed organization of the ternary BLE, the horizontal and vertical interconnect and a switch-box into part of an *island-style* structure. The 3-LUT selects one of 27 ternary values from its configuration latches (note that for compactness Figs. 9 and 10 use the symbol $\bar{1}$ for -1). In common with many commercial FPGA organizations, the LUT can be partitioned, in this case divided into three sub-blocks each terminated with a flip-flop and controlled by I_{1-0} so that each sub-block is approximately equivalent to a binary 3-LUT. This is achieved simply by grounding the input to the (disconnected) input I_3 so that the final stage 3MUX selects the center LUT sub-block. Pass-gate multiplexors allow the flip-flop contents to be derived from the LUT or from the adjacent flip-flop and for each flip-flop to be bypassed in the conventional manner shown previously in Fig. 8. Similarly, it is expected that conventional switch-box topologies [20], [21] will be applicable here, with the proviso that they need to be built using transmission gates rather than the more typical NMOS pass transistor style.

The basic operation of the FPGA topology was evaluated



(a) Simplified interconnect model



(b) Overall performance of one signal path

Fig. 11. FPGA interconnect evaluation example

using the setup shown in Fig. 11a. Here, two BLE blocks are connected through two 500µm lengths of interconnect in metal 1 (M1), linked via a switch-box. This is considered to be a worse-case as it is more likely that one of the orthogonal interconnect lines will be run in metal 2, which exhibits much lower capacitance per unit length than M1. Each BLE connects to a line via a fixed transmission gate as shown, which adds approximately 20pS to the overall propagation delay across the line. The blocks are run from a common clock with zero skew. One path was simulated from source to destination by applying a ternary sequence to T_O and the monitoring the output of the destination flip-flop. The LUT contents were configured such that the output simply followed I_0 (i.e., addresses $\{-1, 0, 1\}$ contain $\{-1, 0, 1\}$). The results shown in Fig. 11b indicate that, as expected the flip-flops $T_O(1)$ and $T_O(2)$ follow the input T_{in} delayed by one clock cycle. The simulated delay from node $T_O(1)$ to the input of the destination flip-flop (Table II) is around 1.24ns worse-case.

D. FPGA Configuration

The ternary configuration register circuit takes advantage of the two-stage organization of the ternary buffer to create a simple quasi-dynamic shift-register function that occupies less area than its corresponding static flip-flop. Based on two

TABLE II
COMPONENT DELAYS FOR MODEL OF FIG. 11A (pS)

Logic Transition	FF(1) clk→Q	t _p Γ _{connect}	t _p 3-LUT	FF(2) clk→Q	Path Delay*
-1→+1	380	400	420	320	1580
+1→-1	320	240	540	280	1420
-1→0	360	300	560	380	1580
0→+1	290	520	360	250	1460
+1→0	220	300	560	300	1300
0→-1	300	400	300	300	1300

*includes setup time

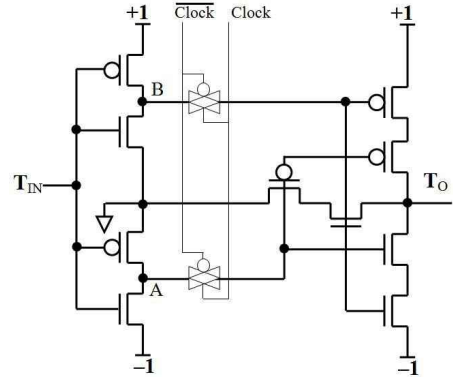
non-overlapping clock signals, (Fig. 12a), the first and second buffer stages are decoupled when clock phase 1 (Ph1) is low. At the same time the latch feedback is disabled (Fig. 12b). Phase 1 (Ph1) is sent high in this period to connect the output stage to the input of the following stage. At the end of the shift cycle, Ph1 returns high and the latch stabilizes around its new value. In this way, a conventional scan-path can be created to write the LUT contents. The inefficiency of the non-overlapping clock mechanism can be justified in this case as it is used only during the configuration process. However, as shown in Fig. 13, the worse-case simulated clock to output delay (FO-4) is approximately 410pS so a configuration clock frequency in excess of 1GHz is conceivable in this technology (assuming that the external programming environment could keep up with this rate).

We have also begun to explore an alternative configuration mechanism that uses an electrically programmable and erasable non-volatile memory element available in the standard SoS process. This EEPROM-like cell comprises a pair of cross-coupled NMOS and PMOS transistors sharing a common channel and a floating gate. In a typical (e.g., radiation hard) application, the cell is programmed negative or positive with respect to ground by injecting either electrons or holes onto the floating gate through the standard gate oxide. Although large compared with existing FLASH cells, it is certainly much smaller than the equivalent clocked latch circuit and has the important advantage of non-volatility. If it proves possible to set up a neutral or un-programmed state around 0V to represent the middle ternary logic state, the cell would represent an ideal building block for the ternary LUT. This will require further experimental work on the cell programming mechanism and will be reported separately.

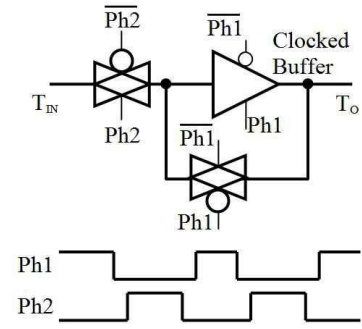
IV. EXAMPLE APPLICATION

In this section we briefly present an application mapped to the ternary FPGA. The objective here is to illustrate in general terms the mapping of an application onto the FPGA. For the moment this is entirely a manual process—in future work we intend to explore design automation techniques using a multi-valued tool flow (e.g., MVSIS [22]) along with issues to do with optimizing the circuit topology to support efficient place and route.

The example used here is the ternary FIR filter with the general form shown in Fig. 14 [5]. The ternary filter output



(a) Clocked buffer circuit



(b) Clocked latch with 2-phase non-overlapping clock

Fig. 12. Ternary LUT configuration shift register cells

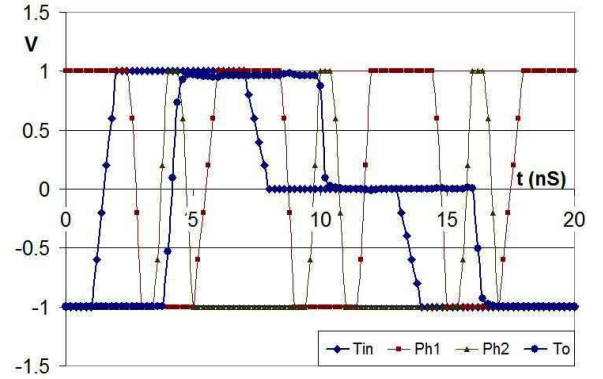


Fig. 13. Clocked latch performance

$y(k)$ is given by the convolution of the sampled input signal x_i with the ternary tap coefficients h_i such that:

$$y(k) = \sum_{i=0}^M h_i x_{k-i} \text{ with } x_i, h_i \in \{1, 0, -1\} \quad (2)$$

where M is the order of the filter. In this case, the input samples are assumed to be derived from a ternary $\Sigma\Delta$ ADC while the coefficients are derived using the $\Sigma\Delta$ modulation of the impulse response of the target filter. The key advantage

here is that the coefficient multiplication stage in Fig. 14 reduces to the selection of a sample, its (ternary) inverse or zero. In this example, the addition is implemented as a tree of 3-input ternary full-adder blocks (Fig. 15). If the filter coefficients are fixed, the multiplication can be merged with the early adder stages and collapsed into one LUT-pair, requiring nothing more than a rearrangement of the terms in Table III. All taps with zero coefficients can be simply removed from the tree. It is assumed here that the coefficients will be supplied from an external source and that these optimizations are not available, so we will be comparing worse case organizations.

In this example, the tree is arranged with $\log_3 M$ stages where the bit lengths increase by one (i.e., the range of the result increases $3\times$) at each stage. In the speech filter of [5], M can be large: 1024 for an over-sampling rate (OSR) of 32, or 2048 for an OSR of 64. In a binary implementation, the coefficient addition would require 10 or 11 stages while its ternary equivalent will comprise 6 or 7 stages.

Table IV compares the approximate area and delay for the ternary FIR filter with an approximately equivalent system implemented on two commercial FPGA devices using a 2-bit binary-coded ternary representation. The two FPGAs were chosen as indicative of low-cost (the 90nm Stratix II) and high-performance versions (the 40nm Stratix IV). All these examples used a simple ripple-carry structure and each was fully pipelined, with a register positioned at the input, output and between each stage, as indicated in Fig. 15. Thus the critical path in each case is the ripple-carry path in the final stage adder, across 11 bits in the binary versions and 7 trits for the ternary case.

Further, we note that the 3LUT topology clearly results in a sub-optimal mapping for the adder components (Fig. 16a), forcing the adder structure to be sub-divided into a number sub-structures mapped across a number of separate LUTs (Fig. 16b). A 4-trit lookup table would directly support (3,2) trinary counters (i.e., 3-input full adders) that would significantly reduce the LUT count, at the cost of additional propagation delay through each. The analysis of these types of optimizations will be the subject of future work.

It must be stressed that the figures shown in Table IV are intended to be indicative only as it is difficult to directly compare the area of these widely divergent technology choices and the ultimate performance of each will be highly dependent on the quality of the fitting, place and route achieved by the tools along with the constraints imposed on the design. Notwithstanding, we can see that the balanced ternary representation results in more compact mappings (fewer, albeit physically larger, LUTs). Although no real attempt has been made at circuit optimization to date, it appears likely that the ternary system implemented on the SoS process will be capable to supporting clock speeds in excess of 233MHz, readily supporting standard video bandwidths at an OSR of 32.

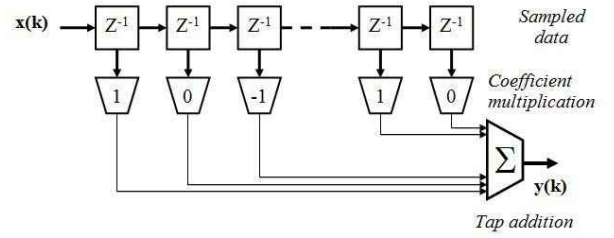


Fig. 14. Ternary FIR Filter Structure (from [5])

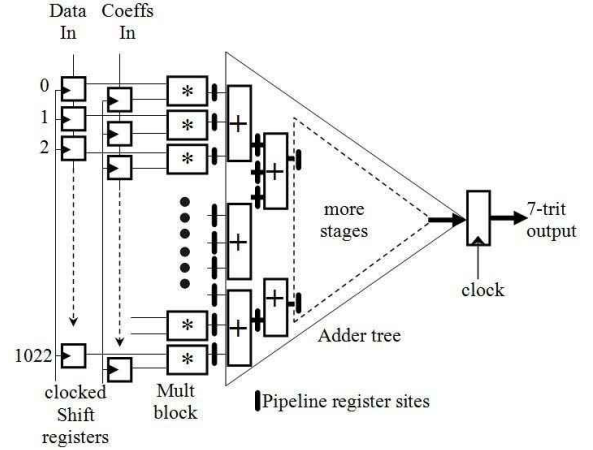
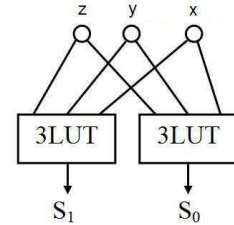
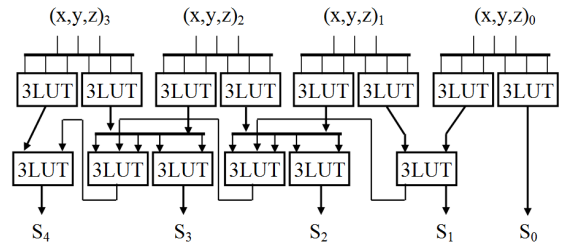


Fig. 15. Ternary FIR Mapped to 3LUT Array



(a) Ternary Half-Adder



(b) 4-Trit 3-Input Ripple-carry Adder using (2,1), (2,2) & (3,1) Counters

Fig. 16. Ternary Adder Stages Mapped to 3LUT Array

TABLE III
TRUTH TABLE FOR TERNARY ADDITION

Input		Output		
x	y	S ₁	S ₀	Decimal
1	1	1	1	+2
1	0	0	1	+1
1	1	0	0	0
0	0	0	0	0
0	1	0	1	+1
0	1	0	1	-1
1	1	0	0	0
1	0	0	1	-1
1	1	1	1	-2

TABLE IV
WORSE-CASE AREA-DELAY ESTIMATES FOR TERNARY FIR FILTER OF FIG. 14

Implementation	Area (# LUTs)	F _{MAX} (MHz)
2's complement binary Cyclone II (90nm)	5119	179.0
Stratix IV (40nm)	5119	617.7
Ternary (250nm 3LUT)	3087	233.6

V. CONCLUSION

In this paper, we have described number of circuit designs that represents form components of the ternary FPGA, including threshold detection, latches, flip-flops and a ternary logic element topology. These designs support a direct implementation of a reconfigurable array structure using a balanced ternary logic system where the logic set $\{\pm 1, 0\}$ maps directly to the equivalent voltage levels, $\{\pm 1.0V, 0.0V\}$. The performance of the ternary components have been simulated and verified using models from a commercially available Silicon-on-Sapphire process.

The key feature of this SoS process that makes it applicable here is the availability within the standard process of multiple transistor threshold voltages. Further, insulated substrate processes are well suited to the integration of RF and analog circuits into a full system on chip. We envisage that the ternary FPGA described here may form just one component of a high-performance DSP platform in which all data and coefficients are represented in balanced ternary format.

At this stage, only the component designs have been completed and their behavior simulated. For this work to proceed, we will need to develop a more complete architectural model of the platform along with a multi-valued CAD flow that will allow us to more completely analyze the architectural tradeoffs in this balanced ternary array.

REFERENCES

[1] D. E. Knuth, *The Art of Computer Programming*. Reading, Mass.: Addison-Wesley Publishing Company, 1981, vol. 2.

[2] P. Wong, "Fully sigma-delta modulation encoded FIR filters," *IEEE Transactions on Signal Processing*, vol. 40, no. 6, pp. 1605–1610, Jun 1992.

[3] G. Frieder and C. Luk, "Algorithms for binary coded balanced and ordinary ternary operations," *IEEE Transactions on Computers*, vol. C-24, no. 2, pp. 212–215, Feb. 1975.

[4] Y. Iguchi, M. Matsuura, T. Sasao, and A. Iseno, "Realization of regular ternary logic functions using double-rail logic," in *Proceedings of the Asia and South Pacific Design Automation Conference, ASP-DAC'99*, Jan 1999, pp. 331–334 vol.1.

[5] A. Thompson, P. O'Shea, Z. Hussain, and B. Steele, "Efficient single-bit ternary digital filtering using sigma-delta modulator," *IEEE Signal Processing Letters*, vol. 11, no. 2, pp. 164–166, Feb. 2004.

[6] A. C. Thompson, Z. M. Hussain, and P. O'Shea, "A single-bit narrow-band bandpass digital filter," *Australian Journal of Electrical and Electronics Engineering*, vol. 2, no. 1, pp. 31–40, 2005.

[7] Z. Sadik, Z. M. Hussain, and P. O'Shea, "An adaptive algorithm for ternary filtering," *IEE Electronics Letters*, vol. 42, no. 7, pp. 420–421, March 2006.

[8] Sapphicon Semiconductor. [Online]. Available: <http://www.sapphicon.com/>

[9] The mosis service. [Online]. Available: <http://www.mosis.com/>

[10] T. Rajashekhara and I.-S. Chen, "A fast adder design using signed-digit numbers and ternary logic," in *Proceedings of the 1990 IEEE Southern Tier Technical Conference*, Apr 1990, pp. 187–194.

[11] A. Srivastava and K. Venkatapathy, "Design and implementation of a low power ternary full adder," *VLSI Design*, vol. 4, no. 1, pp. 75–81, 1996.

[12] T. Shibata and T. Ohmi, "A functional MOS transistor featuring gate-level weighted sum and threshold operations," *IEEE Transactions on Electron devices*, vol. 39, no. 6, pp. 1444–1455, 1992.

[13] H. Gundersen and Y. Berg, "A novel balanced ternary adder using recharged semi-floating gate devices," in *ISMVL '06: Proceedings of the 36th International Symposium on Multiple-Valued Logic*. Washington, DC, USA: IEEE Computer Society, 2006, p. 18.

[14] T. Felicijan and S. Furber, "An asynchronous ternary logic signaling system," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 11, no. 6, pp. 1114–1119, Dec. 2003.

[15] J.-M. Philippe, E. Kinvi-Boh, S. Pillement, and O. Sentieys, "An energy-efficient ternary interconnection link for asynchronous systems," in *Proceedings of the IEEE International Symposium on Circuits and Systems, ISCAS 2006*, 2006, pp. 1011–1014.

[16] C. Duan and S. P. Khatri, "Energy efficient and high speed on-chip ternary bus," in *DATE '08: Proceedings of the conference on Design, automation and test in Europe*. New York, NY, USA: ACM, 2008, pp. 515–518.

[17] E. Sipos, L. Festila, and G. Oltean, *Towards Reconfigurable Circuits Based on Ternary Controlled Analog Multiplexers/Demultiplexers*, ser. Lecture Notes in Computer Science: Knowledge-Based Intelligent Information and Engineering Systems, I. Lovrek, R. Howlett, and L. Jain, Eds. Springer-Verlag Berlin Heidelberg, 2008, vol. 5179.

[18] S.-M. Kang and Y. Leblebici, *CMOS Digital Integrated Circuits: Analysis and Design*. McGraw-Hill, 1996.

[19] V. Betz and J. Rose, "How much logic should go in an FPGA logic block?" *IEEE Design and Test of Computers*, vol. 15, no. 1, pp. 10–15, January-March 1998.

[20] H. Fan, J. Liu, Y.-L. Wu, and C.-C. Cheung, "On optimum switch box designs for 2-D FPGAs," in *Proceedings of the Design Automation Conference, DAC 2001.*, 2001, pp. 203–208.

[21] H. Fan and Y.-L. Wu, "Crossbar based design schemes for switch boxes and programmable interconnection networks," in *Proceedings of the Asia and South Pacific Design Automation Conference, ASP-DAC 2005*, vol. 2, Jan. 2005, pp. 910–915 Vol. 2.

[22] R. Brayton and S. Khatri, "Multi-valued logic synthesis," in *Proceedings of the Twelfth International Conference On VLSI Design*, Jan 1999, pp. 196–205.