An Efficient Reducing Trasient Activity on Off-Chip Data Bus With Dynamically Varying Switching Characteristics

G.Sangeetha[1], Dr.S.Vijayakumar[2], S.Thiruvasagam[3]
Department of Electronics and Communication Engineering
Paavai Engineering College, Namakkal

Abstract-*Power has become an important design criterion in modern system designs, especially in portable battery-driven applications. A significant portion of total power dissipation is due to the transitions on the off-chip address buses. This is because of the large switching capacitances associated with these bus lines. There are many encoding schemes in the literature that achieve huge reduction in transition activity on the instruction address bus. However, on data and multiplexed address buses, none of the existing schemes consistently achieve significant reduction in transition activity. Also, many of the existing techniques add redundancy in space and/or time. The encoding schemes are proposed that significantly reduce transitions on these buses without adding redundancy in space or time. The TPC algorithm provides the optimal bus encoding, given the probabilistic distribution of the data. The shortcomings are that its effectiveness relies on the accurate information about the data sequence, and the hardware and computation overheads are high and increase exponentially with the increase of bus width. We propose a novel low-energy bus coding scheme. The proposed weighted code mapping (WCM) algorithm generates a one-to-one mapping from the original data to the coded data, which is decided by the probabilistic distribution of the original data.*

Keywords: Bus encoding schemes, encoder–decoder architecture, low power VLSI designs, off-chip bus.

I. INTRODUCTION

With the increase of speed and complexity of today's designs due to need for increased performance and the demand, there is significant increase in the power consumption of VLSI chips. Power optimization has become a serious concern for achieving high reliability and low packaging cost. A substantial fraction of the total energy consumption comes from system buses [1], [2], because the capacitance associated with an external pin is usually much larger (up to three orders of magnitude) than that of the internal nodes [3], [4]. Therefore, researchers have explored many techniques for minimizing switching activity at external high capacitance off-chip buses at the expense of additional transitions on internal low capacitance nodes. Off-chip bus power minimization can be done by reducing the supply voltage, buscapacitance, or the total number of signal transitions in each bit line.

In this paper, we are focusing on reducing transitional activity on data streams whose statistics are not known a prior and switching characteristics change spatially (across thewidth of the bus) and temporally. We propose an adaptive clustering technique by observing data over time with the aidof spatiotemporal redundancy. In each observation window, we extract local switching statistics and form a subgroup, i.e., aset of bit lines that has highly correlated switching pattern inthat observation window.

Bus encoding schemes are techniques that reduce the transitional count, which further enhances the energy savings. Several groups utilize spatial redundancy [12], [19] in the form of extra bus lines while others exploit temporal redundancy [15]in the form of excess bit transmission to develop transition eduction encoding mechanism..

A.PROPOSED ENCODER AND DECODER ARCHITECTURES

In this section, we will demonstrate possible implementation of the proposed algorithm. Fig. 1 shows basic block diagram of proposed encoding methodology for bus of width Nit consists of decision blocks, delay elements, and set of XOR gates and a multiplexer. Decision block consists of eliminator, cluster formation, and basis selection unit. It generates the control information, corresponding to cluster for each observation window and the multiplexer inserts the temporal redundancy.

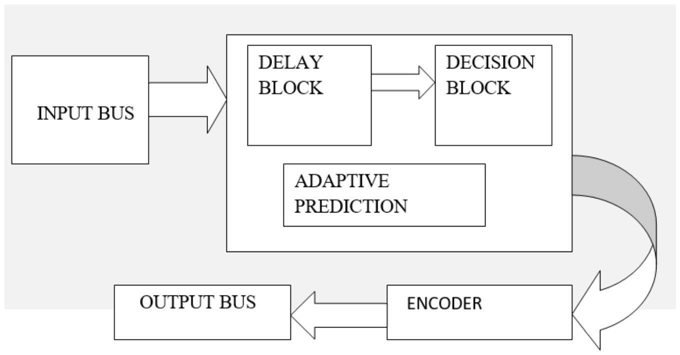

Figure1 Block -level encoder architecture of proposed methodology

The decoder architecture is presented in Fig. 2 which retrieves the data back to its original form. The cluster information is sent as control signal at the beginning of encoded data for every observation window. Before decoding the data, the decoder extracts this cluster information by observing the transition between control signal and encoded data of previous observation window. This information is kept in the register for each bit line until the end of decoding for current window.

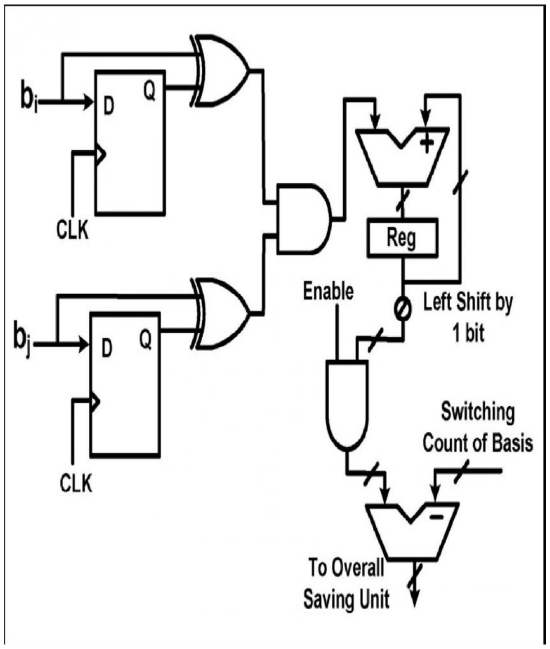

Figure 2 Decoder architecture of proposed scheme

TABLE I
POWER CONSUMPTION AND AREA OVERHEADFOR ENCODER AND DECODE

Block	Area (in sq µv)	Area(NAND2 Equivalent)	Static power (in µv)	Dynamic power (in µv)
Encoder	162K	12.8K	9461	20293
Decoder	1.9K	120	141	0.024

B. ALGORITHM

The high-level description of the procedure for every observation Window can be written as follows

```
1) Find total number ($s_{\text{ini}}$) of switching transitions in all the bit lines
 2) For $i = 0$ to $(N - 1)$
        a) Choose $i$ th line as basis
        b) For $j = 0$ to $(N - 1), j \_= i$
i) Put $j$ th line in cluster if $j$ th line has more switching transitions than $j$ th line XOR-ed with basis
line [see (4)]
c) end
d) XOR all the clustered bit lines with the basis line
e) Find number ($s_i$) of switching transitions in this modified set
f) Savings $(\rho_i) = s_{\text{ini}} - s_i$[alternate method of computing (5)]
3) end
4) If $b_k = \text{argmax}_i(\rho_i)$, $0 \le i \le (N - 1)$, then the $k$th bit-line is the basis line for that observation
window [see (6)]
```

Basis and cluster information is sent to receiver by one extra bit line and using extra time slot. Extra bit line does not increase any switching transitions, and insertion of control information between two observation windows has minimal effect on the number of switching transitions. This is elaborated in Section V-B along with the selection of basis in each observation window.

II. FPGA DESIGN IMPLEMENTATION (XILINX DESIGN FLOW)

1-CORE Technologies provide FPGA design service of high quality since 2004. Outsourcing FPGA design to Russia will significantly reduce your design costs. In this chapter of the FPGA design tutorial Xilinx implementation flow is discussed in details.

Xilinx implementation flow

Xilinx ISE has three implementation steps: translate, map and place and route. These steps are specific for Xilinx for example, Altera combines translate and map into one step executed by quartos map.

Translate

Translate is performed by the NGDBUILD program. During the translate phase an NGC netlist (an output of a synthesizer) is converted to an NGD net list. The difference between them is in that NGC netlist is built around UNISIM component library, designed for behavioral simulation, and NGD netlist is dependent upon the SIMPRIM library. The NGDBUILD-produced net list therefore contains some initial information about switching delays (but it is approximate).

MAP

Mapping is performed by the MAP program. During the map phase the SIMPRIM primitives from an NGD netlist are mapped on specific device resources: LUTs, flip-flops, BRAMs and other. The output of the MAP program is stored in the NCD format. In contains precise information about switching delays, but no information about propagation delays (since the layout hasn't been processed yet. For Virtex-5 devices MAP also does placement (see below). For other devices placement is done by PAR. Routing is done by PAR for all devices.

Place and route

Placement and routing is performed by the PAR program. Place and route is the most important and time consuming step of the implementation. It defines how device resources are located and interconnected inside an FPGA. Placement is even more important than routing, because bad placement would make good routing impossible. In order to provide possibility for FPGA designers to tweak placement, PAR has a "starting cost table" option. PAR accounts for timing constraints set up by the FPGA designer. If at least one constraint can't be met, PAR returns an error. The output of the PAR program is also stored in the NCD format. For Virtex-5 devices, placement is performed by MAP instead of PAR.

Timing constraints

In order to ensure that no timing violation (like period, setup or hold violation) will occur in the working design, timing constraints must be specified. Basic timing constraints that should be defined include frequency (period) specification and setup/hold times for input and output pads. The first is done with the PERIOD constraint, the second - with the OFFSET constraint. Timing constraints for the FPGA project are defined in the UCF file. Instead of editing the UCF file directly, an FPGA designer may prefer to use an appropriate GUI tool. However, the first approach is more powerful.

Specifying timing groups

Timing groups are sets of objects to which constraints should by applied. There are some pre-defined groups, and more groups can be created by an FPGA designer.

There are many forms of timing constraints. Some of them will be discussed below, other can be found in Xilinx Constraints Guide. One way to create a timing group is to link it with the particular net:
NET "net_name" TNM_NET = qualifier "tnm_name";

Here net_name is a name of clock net (signal), tnm_name is a name of the associated timing group, and qualifier is an optional condition which is used to include in the group only elements of particular types. Examples of qualifiers include FFS (for flip-flops), PADS (for pads), RAMS (for RAM modules).

TNM_NET example:
NET "CLK" TNM_NET = "clk_net";
This code defines a clk_net timing group associated with the CLK clock net and including all synchronous elements controlled by this net (since no qualifier has been specified).
Another way to define a timing group is to specify the name of the instance (module):
INST "inst_name" TNM = qualifier "tnm_name";
In this case all synchronous elements in inst_name (or those corresponding to qualifier when specified) are grouped in tnm_name group.

Specifying period

PERIOD constraint is used to define a clock period (or frequency) for the clock domain. Example:
NET "CLK" TNM_NET = "clk_net"; TIMESPEC "TS_clk_net" = PERIOD "clk_net" 10 ns;
This example defines a minimum period of 10 ns for CLK clock net (which corresponds to 100 MHz frequency). TS_clk_net is a constraint name, it can be any other.

Specifying offset

OFFSET constraint is used to specify external setup time for input pads or necessary hold time for output pads. For input pads, OFFSET specifies a time before the (external) clock edge when the related data signals are set. Example:
OFFSET = IN 5 ns BEFORE "CLK";
For output pads, OFFSET specifies a minimum time after the clock edge when the related data signals can be deserted. Example:
OFFSET = OUT 2 ns AFTER "CLK";

In this form, constraints are applied to all I/O pads related to CLK clock signal. There are also more specific timing constraints. For more information, see Xilinx Constraints Guide.

False paths

A false path is a path between two synchronous elements which formally exists, but by design won't be ever activated during the device operation. As such, false paths should be excluded from static timing analysis. It is done using the TIG constraint:

TIMESPEC "TSid" = FROM "from_grp" TO "to_grp" TIG;

Here TSid is an identifier of the constraint, and from_grp and to_grp are timing groups.

Multicycle paths

There are situations when a given combinational logic unit is designed to produce output after more than one clock cycles. Such paths must be defined as multicycle; otherwise implementation tools will try to fit them in one cycle, possible failing the place and route.

Example: imagine that we have a TS_clk_net TIMESPEC constraint for clock period. Then in order to define a multicycle path from the multi_start group to the multi_end group we should write:
TIMESPEC "TS_multicycle_01" = FROM "multi_start" TO "multi_end" TS_clk_net*2;

I/O pads assignment

I/O pads constraints should be specified in any real design. In the absence of these constraints the implementation tools will choose pads on its own. It is not that an FPGA designer usually need. I/O pad constraints can be conveniently set up with the Assign Package Pins utility from Xilinx ISE.The example of an UCF syntax for I/O pads constraints is NET "data_input" LOC = AK14 | IOSTANDARD = LVCMOS33 | SLEW = SLOW;LOC

Constraint specifies a pin number (AK14). IOSTANDARD constraint specifies I/O standard (low-voltage CMOS 3.3V). SLEW constraint specifies slew rate (slope steepness). It is recommended to use SLOW slew rate whenever possible, since it reduces EMI (electromagnetic interference). There are also other parameters that can be set for I/O pads.

Xilinx ISE (Integrated Software Environment) is a software tool produced by Xilinx for synthesis and analysis of HDL designs, enabling the developer to synthesize ("compile") their designs, perform timing analysis, examine RTL diagrams, simulate a design's reaction to different stimuli, and configure the target device with the programmer.

The Web Edition is a free version of Xilinx ISE that can be downloaded at no charge. It provides synthesis and programming for a limited number of Xilinx devices. In particular, devices with a large number of I/O pins and large gate matrices are disabled.

The low-cost Spartan family of FPGAs is fully supported by this edition, as well as the family of CPLD’s, meaning small developers and educational institutions have no overheads from the cost of development software. License registration is required to use the Web Edition of Xilinx ISE, which is free and can be renewed an unlimited number of times.

III. CONCLUSION

In the conclusion, a new adaptive bus encoding techniqueis proposed, which does not require prior knowledge aboutthe signal characteristics. The proposed algorithm selectively encodes a cluster of lines, which have maximum correlated switching transitions within the fixed observation window, by performing XOR operation with a basis bit line, selected based on the switching characteristics of the bit lines in that window. Hence, the clustering is adaptive. The area overhead and the power consumption of the encoder and decoder architectures are studied in Synopsys using 180-nm technology file. The critical path of the overall block involves delay for only one such unit (for the MSB) and ANDgate delays for the remaining bits. Thus, this implementation of the comparator is found to be more efficient than the conventional comparator in terms of power consumption and critical pathdelay.

REFERENCES

[1] H. B. Bakuglo, Circuits, Interconnections and Packaging for VLSI.Menlo Park, CA, USA: Addison-Wesley,1990.
[2] T. Burd and B. Peters, “A power analysis of a microprocessor: A study of an implementation of the MIPS R3000 architecture,”Univ. California Berkeley, Berkeley, CA, USA, Tech. Rep., May 1994.
[3] M. Stan and W. P. Burleson, “Limited-weight codes for low-power,”inProc. IEEE/ACM Int. Workshop LowPower Design (IWLPD),Napa Valley, CA, USA, Apr. 1994, pp. 209–214.

[4] M. R. Stan and W. P. Burleson, “Low-power encodings for global communication in CMOS VLSI,” IEEE Trans. Very Large Scale Integer.(VLSI) Syst., vol. 5, no. 4, pp. 444–455, Dec. 19977.
[5] C. L. Su, C. Y. Tsui, and A. M. Despain, “Saving power in the control path of embedded processors,” IEEE Design Test Comput., vol. 11, no. 4,pp. 24–30, Dec. 1994.
[6] L. Benini, G. De Micheli, E. Macii, D. Sciuto, and C. Silvano,“Asymptotic zero-transition activity encoding for address busses inlow-power microprocessor-based systems,” in Proc. IEEE Great LakesSymp. (GLS-VLSI), Urbana, IL, USA, Mar. 1997, pp. 77–82.
[7] Lei Zhang, Xiao-jingGuo, Xiao-Pei Wu, (2013) “Low-cost Circuit Design of EEG Signal Acquisition for the Brain-computer Interface System”.IEEE 6th International Conference on Biomedical Engineering and Informatics, pp. 245-250.
[8] D. J .A. Groeneveld, “Bandwidth extension and noise cancelling for TIAs”, MScThesis, University of Twente, September 2010.
[9] C.C. Sthalekar, and V.M.J. Koomson, "A CMOS Sensor for Measurement of Cerebral Optical Coefficients”. using Non-Invasive Frequency Domain near Infrared Spectroscopy,IEEE Sensors Journal, no.99, pp.1, 2013.
[10] K. Foubert, et al. "Development of HgCdTe single-element APDs based detectors for low flux short waveinfrared applications”.SPIE OPTO. Informational Society for Optics and Photonics, 2013.
[11] J.-S. Youn, et al. "An integrated 12.5-Gb/s optoelectronic receiver with a silicon avalanche photodetector in standard SiGe BiCMOS technology.".Optics expresses, 20.27, 28153-28162, 2012.
[12] S.-H. Huang, W.-Z. Chen, Y.-W. Chang and Y.-T.Huang, “A 10-Gb/s OEIC with meshed spatially-modulated photo detector in 0.18-μm CMOS technology”.IEEE J. Solid State Circuits,46(5),1158–1169,2011.
[13] J.-S. Youn, M.-J.Lee, K.-Y. Park and W.-Y. Choi, “10-Gb/s 850-nm CMOS OEIC receiver with a silicon avalanche photodetector”.IEEE J. Quantum Electron.48(2), 229–236, 2012.