

# Design for an IO block array in a tile-based FPGA

Ding Guangxin(丁光新)<sup>†</sup>, Chen Lingdou(陈陵都), and Liu Zhongli(刘忠立)

(Institute of Semiconductors, Chinese Academy of Sciences, Beijing 100083, China)

**Abstract:** A design for an IO block array in a tile-based FPGA is presented. Corresponding with the characteristics of the FPGA, each IO cell is composed of a signal path, local routing pool and configurable input/output buffers. Shared programmable registers in the signal path can be configured for the function of JTAG, without specific boundary scan registers/latches, saving layout area. The local routing pool increases the flexibility of routing and the routability of the whole FPGA. An auxiliary power supply is adopted to increase the performance of the IO buffers at different configured IO standards. The organization of the IO block array is described in an architecture description file, from which the array layout can be accomplished through use of an automated layout assembly tool. This design strategy facilitates the design of FPGAs with different capacities or architectures in an FPGA family series. The bond-out schemes of the same FPGA chip in different packages are also considered. The layout is based on SMIC 0.13  $\mu\text{m}$  logic 1P8M salicide 1.2/2.5 V CMOS technology. Our performance is comparable with commercial SRAM-based FPGAs which use a similar process.

**Key words:** FPGA; IO block; signal path; configurable IO buffer; layout; packaging

**DOI:** 10.1088/1674-4926/30/8/085008

**EEACC:** 2570A

## 1. Introduction

Configurability is the key reason why an FPGA can be an efficient ASIC alternative in designing an electronic system. When choosing an FPGA for use in the target application, one critical consideration is whether the mapping of the application input-output (IO) pins to the IO cells of the FPGA can fulfill the needs of the input/output interface in functionality and performance. Thus when planning for the IO design of an FPGA family, all the IO cells are designed by taking into account not only the potential application conditions but also the different IO cell organizations in the individual FPGA family members. The design strategy is to keep a consistent functionality and performance specification throughout all members of the FPGA family. Island style is a popular FPGA structure in some commercial products<sup>[1,2]</sup> and much research work has been done on this<sup>[3,4]</sup>. In island-style FPGA, the configurable logic blocks (LBs) are surrounded by generic routing resources which are arranged in rows and columns. The LB and its adjacent routing channels together are treated as a basic design tile that can be repeated and replaced in the FPGA composition. Taking advantage of this structure, a tile-based methodology is used to automate the FPGA design, verification and layout<sup>[5]</sup>. This methodology requires a hierarchical design strategy. As a result, the consideration extends from the tile-based IO cell (IOC, corresponding to an IO pin) design to the construction of the IO block (IOB, composed of typically two to four IOCs) and IOB array design. The design of individual IOCs not only has to consider all different mode selections for use in different IO standards but also all possible IOC organizations in the IOB array for signal routing. The programming and boundary scan logic are all supported

in the IOB array design. The IOB array design also needs to consider different bond-out options in different packages of the same FPGA chip. The high performance requirement and high degree of functionality integration to accommodate different applications of the FPGA make the design of individual IOCs and the IOB array in all members of the FPGA family a very complex and challenging task. In this paper, we intend to present our effort and result in accomplishing this task through the design of both individual IOC circuitry and the coordination of the IOC in the entire IOB array.

The overall architecture of the IOB array, which includes the classification of all different IOC types used in the target applications of the FPGA family, is described. The circuitry of a typical IOB and its IOC components is described. The IOB array layout integration and the overall performance are described. The design parameters of the IOB array are based on SMIC 0.13  $\mu\text{m}$ , 1.2 V core voltage, 2.5 V IO voltage, 1P8M CMOS technology.

## 2. Overall architecture of IOB array

As the basic components of the IOB array, IOCs can be classified into a number of types according to their functionalities. Four types of IOCs are described as follows:

### (1) User IO

This is a general-purpose user-defined input/output data interface when FPGA is in the operation mode. The majority of IOCs in the IOB array belong to this type. It provides versatile multiple modes for the user to configure in each application of the FPGA. Each individual user IO can be configured to comply with a special input/output standard for an application. When used in a differential IO standard, the user-IO IOCs

<sup>†</sup> Corresponding author. Email: gxding@red.semi.ac.cn

Received 9 March 2009, revised manuscript received 6 April 2009

are paired to form a differential IO pair.

(2) Dual-purpose IO

This type of IOC can work as either a general-purpose user IO or a special-function IO. These IOs include some configuration pins, reference voltage supply pins, global clock pins, etc. Take the global clock pins as an example. In addition to being connected to the routing channel resources, the output of a global clock IOC is also connected to a separate clock tree routing resource, where global clock signals are generated and distributed throughout the entire chip. If the pin is configured as a user IO, the clock-tree-routing path will be turned off, and vice versa.

(3) Dedicated-function IO

This IO only serves a dedicated function for programming or JTAG. Unlike a user IO or dual-purpose IO, it has no configurable resources and can not be configured for other functions. All dedicated-function IOs use their own specific power supply (VAUX, detailed discussion in Section 3.3), differing from the user and dual-purpose IO VCCO power supply.

(4) Power IO

This provides a power supply for input/output buffers (VCCO), internal core logic (VCCINT), ground (VSS), and some specific circuitry (VAUX).

The total number and locations of the IOCs in the IOB array need to consider the IO mapping in the target applications, the matching with the logic capacity in the FPGA core, power supply distribution, the reference voltage used in the applicable IO standards and the constraints from the target packages. In some cases, the IO mapping might need to accommodate different VCCO voltages and reference voltages in the same FPGA. Thus IO banks, which are based on different VCCOs, are introduced. As shown in Fig. 1, the IOB array is divided into eight banks, with two banks on each side. Each bank has a dedicated VCCO and its own reference voltage. The IOCs in each bank share the same VCCO and reference voltage. Each bank can thus support an independent VCCO voltage, which would be 1.5, 1.8, 2.5 or 3.3 V. The dedicated-function IOCs do not belong to any banks, and are located close to the corners of the IOB array. The VCCINT and VSS IOCs are evenly distributed in the IOB array.

The numbers of IOCs in each bank, their types and locations are all specified in the FPGA architecture definition file.

**3. Circuit design of IOB array**

Even though all four types of IOCs are different in their functionalities, their circuit structure can all be divided into two segments: the signal path and the IO buffer.

As shown in Fig.1, the signal path of a typical IOC is composed of user data input-output logic (data path), programming logic (programming path) and boundary-scan logic (boundary-scan path). The data path provides user with some optional functions such as data synchronization and double-data-rate (DDR) transmission. The programming logic processes the signals between the on-chip programming

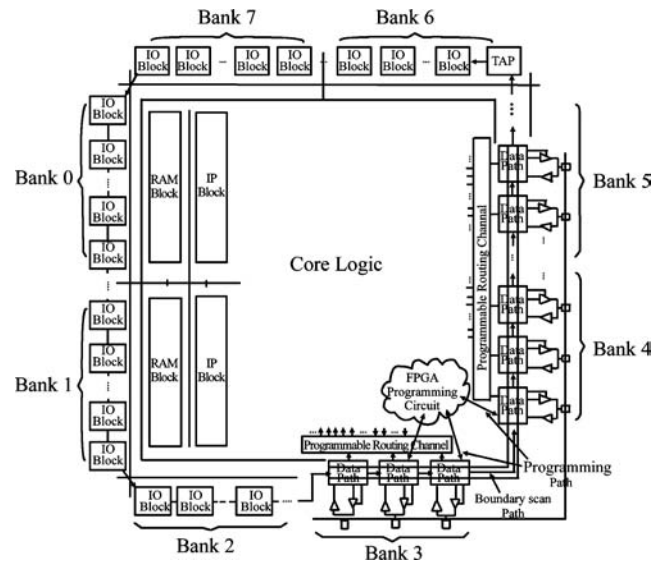


Fig. 1. Simplified diagram of the IOB array.

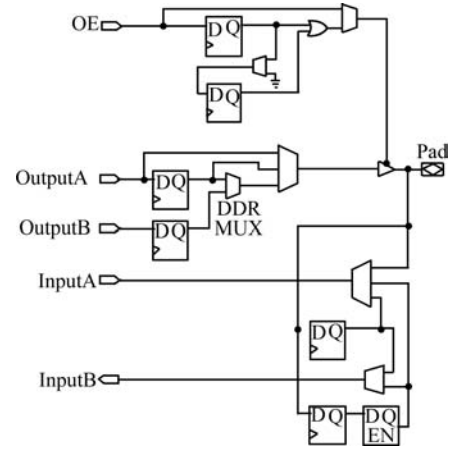


Fig. 2. Structure of a common data path.

circuit and the IO buffer for the programming purpose. This logic only exists in some programming purpose IOCs. The boundary-scan path shifts the serial data into the register chain and loads data into separate associated latches for testing purposes. The local routing pool is a programmable interconnect network between the data path and chip routing channel. It is shared by adjacent IOCs in the same IOB. The configurable IO buffers provide multiple IO standards for both single-ended (e.g., LVCMOS, LVTTTL, PCI, GTL, SSTL, HSTL) and differential signaling (e.g., LVDS).

**3.1. Signal path circuitry**

Figure 2 shows the structure of the data path in a commercial FPGA<sup>[1,2,6]</sup>. The signal path circuitry processes the external signals through three configurable logic circuit paths: the input signal path, the output signal path and the tri-state control signal path. The names follow the convention used in the user data logic. The tri-state input-output logic composed of the three paths is configurable to accommodate different data flows in individual IO standards. Each path contains two configurable registers. The first register in each signal path would shorten the data set-up time (input path) and clock-to-output time (output path) and provides a high data rate and consis-

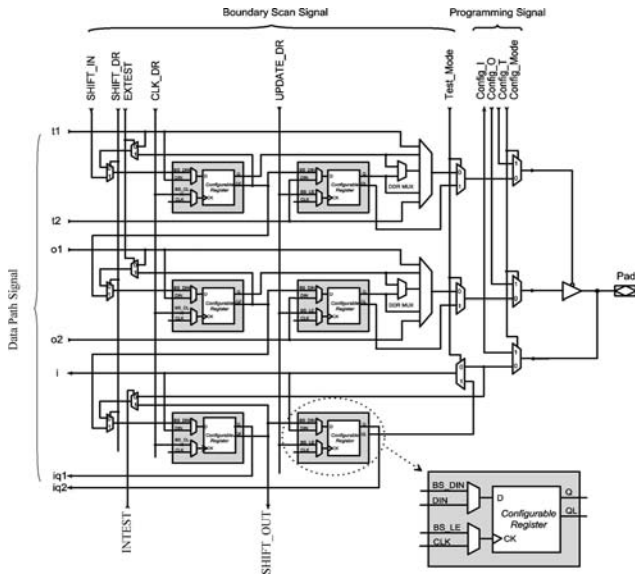


Fig. 3. Simplified signal path circuitry diagram.

tent external IO pin timing (from build to build) in the data transmission applications. Each register can be configured as an edge-triggered D-typed flip-flop or level-sensed latch. Polarities of the clock/latch-enable and set-reset input are also configurable. Coordinating with a phase locked loop (PLL) or delay locked loop (DLL) block embedded in FPGA, the register pair can perform DDR operations.

As the boundary-scan logic is used in FPGAs mainly for programming and some board-level tests<sup>[7]</sup>, we propose a signal path structure with shared registers for both user data input-output and boundary-scan logic. As shown in Fig. 3, through different configurations of the input/output multiplexers, the register pair can be configured to function in user DDR data input/output logic or to work as the store-and-load register pair in the boundary-scan mode. If boundary-scan mode is entered, the six registers are all used for data shifting or data loading. As a result, the data registers used in the input, output and tri-state control data paths need to be mapped to registers in the core logic to avoid data corruption by boundary-scan operation during application. The register sharing can thus be utilized for boundary-scan programming during FPGA configuration as well as boundary-scan testing during normal operation mode. This structure meets most boundary-scan test requirements in normal FPGA applications.

The programming logic bypasses the registers and directly transfers data between the on-chip programming circuit and the IO buffers of individual dedicated programming IOCs.

This circuit structure takes advantage of the existing configurable registers in the data path and avoids the need for designing dedicated boundary-scan registers. As shown in Table 1, the area saved from this structure can be estimated by comparing three different design styles. Register sharing leads to 5% area saving as compared to two dedicated register pairs for both data IO and boundary-scan functions.

**3.2. IOB local routing pool**

In forming the IOB array, the IOCs are grouped into

Table 1. Area comparison for different register styles.

| Design style   | Area ( $\mu\text{m}^2$ ) |
|--|--------------------------|
| Two registers for data IO only                         | 336                      |
| Dedicated register pairs for data IO and boundary scan | 384                      |
| Two registers shared by data IO and boundary scan      | 363                      |

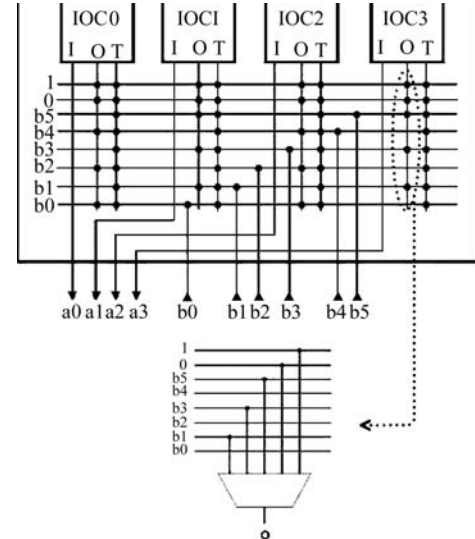


Fig. 4. Structure of the IOB local routing pool.

IOBs. As shown in Fig. 4, the routable signals to/from the IOCs are connected through a shared programmable interconnect network, called the IOB local routing pool, to the external routing channel surrounding the IOB array. The IOB local routing pool is designed to increase IO pin routability for resolving the congestion caused by pin-locking in the FPGA application. The local routing pool also provides programmable switches to connect the IOC inputs to VCCINT or VSS in case of unused or fixed IOC inputs.

Since each IOB is composed of IOCs of different types, an automation tool is used to generate the routing switch pattern and its layout is based on the architectural specification of the target FPGA.

**3.3. Programmable IO buffers and auxiliary power**

In designing the IO buffers to accommodate multiple IO standards, a 3.3 V tolerable 2.5 V IO process is used to support the industry-standard 3.3 V LVTTTL and 2.5/3.3 V LVCMOS standards. However, it is a challenge to support the high-speed and high current drive 1.5 V LVCMOS standard using 3.3 V gate oxide. As a result, an auxiliary supply (VAUX) is introduced. Both VCCO and VAUX supply power for the IO buffer segment, but the output signal swing of the programmable buffer is only determined by VCCO. VAUX is only an auxiliary supply for some specific circuit (level shifter) in the buffer segments. VAUX powers the pre-driver of the output buffer as shown in Fig. 5.

This auxiliary power supply is always set to 2.5 V, so that the control signals (output of the level shifters) of the output buffer swing between VSS and 2.5 V. This provides a fast slew rate and strong drive current as the VGS of the NMOS remains

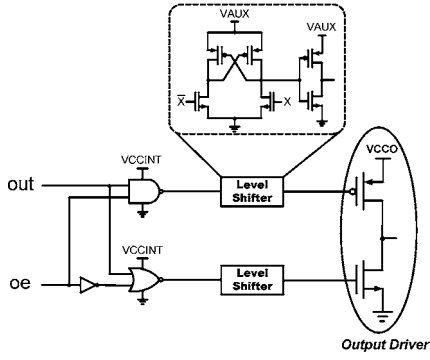


Fig. 5. Block diagram of the output circuit.

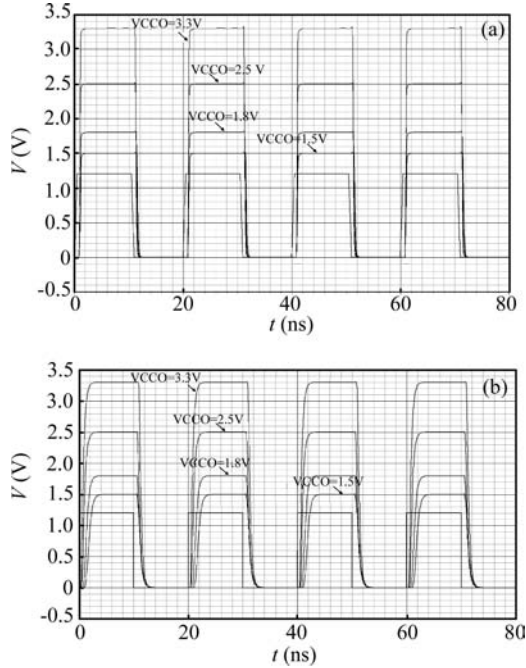


Fig. 6. Waveforms of IO buffer operating LVC MOS: (a) Waveforms with VAUX; (b) Waveforms without VAUX.

at 2.5 V when 1.5 V VCCO is applied. As a result, the pull-down NMOS size can be reduced when using the auxiliary power supply to achieve the same 1.5 V performance without an increase in die area or pin capacitance<sup>[8]</sup>.

Figure 6 shows the performance of the output driver configured for operation in the LVC MOS mode with load of  $R = 50 \Omega$  and  $C = 10 \text{ pF}$  in the typical PVT condition. Figure 6(a) shows when VAUX is applied, and Figure 6(b) is when VCCO, instead of VAUX, is applied to the pre-buffering stage. We can see that VAUX obviously improves the performance of the output driver, especially for the 1.5 V and 1.8 V LVC MOS standards.

## 4. Layout and performance

### 4.1. Power distribution and routing strategy<sup>[9]</sup>

One important consideration in the IOB array design is the power delivery to the core logic. Two issues play an important role: electromigration reliability and DC and AC signal integrity.

The power and ground signals used in the IOB array are

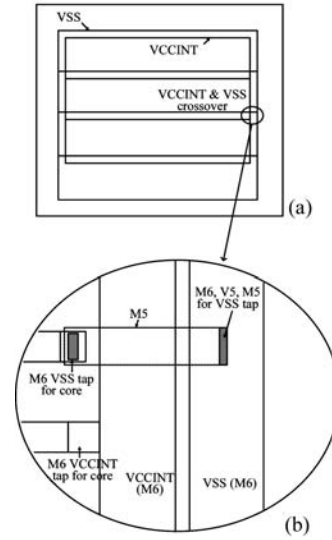


Fig. 7. VCCINT and VSS signal cross-over.

laid out by the topmost metal layers, typically the thickest and providing the lowest resistance. VCCO power delivery is limited within each bank and does not form a ring in the IOB array. It also does not provide the power of the core logic and can be placed close to the outer edge of the IOB array. VCCINT and VSS provide the power of the core logic and each forms a ring close to the inner edge of the IOB array. Both the VSS and VCCINT rings use the top-layer metal M6. The VSS usually has a substrate return assisting it, whereas the VCCINT is completely dependent on the peripheral power delivery from the IOB array. The connection from the VCCINT and VSS rings to the power/ground wires in the core logic forces the two signals to cross over. With proper placement of the power pads, the cross-over points and the resulted cross talk can be minimized.

As shown in Fig. 7, the VSS wire metal is made to cross over VCCINT. In this arrangement, VCCINT and VSS are both in M6. The cross-over metal is at M5 and is tapped to the VSS wire in the IOB array through VIA5. The tapping from the VSS wire in the IOB array to the core VSS is through direct connection on M6. This is a particular implementation of a power grid to illustrate the key concepts. A real power grid will depend on the actual package, chip, clock speed, and voltage and current requirements that may look substantially different.

Another critical step in IOB array design is the allocation of routing channels for the IO signals. Abutting IOC layouts are planned to simplify and reduce routing, especially when our design methods target an FPGA family.

The shift signals between neighbors used in the boundary-scan IOC chain for testing purposes fall into this category. The signal wires can be abutted onto adjacent IOCs and contribute to significant saving in the routing area. The voltage reference signal group is another example of the connection by abutment of the IOCs in the same bank. Other layouts that regularly abut are the power supply, IO buffer mode setting signals, etc. One routing strategy is shown in Fig. 8. The signals in the IO buffer are all local to each IOC and only use

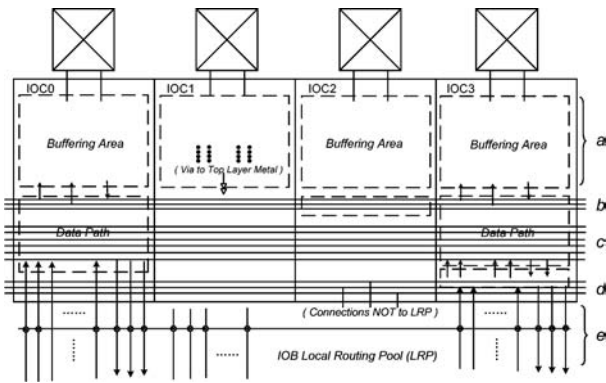


Fig. 8. Layout plan of an IOB.

lower-layer M1, M2 metals. Area *a* is for IOC buffer segments. Power delivery metals (VCCO, VAUX, VSS and VCCINT) are abutted here. Area *b* is a channel for signals used for programming and boundary-scan logic.

Signals used in three signal data paths—input, output and tri-state control signals as well as IO buffer configuration signals are routed in Area *c*. For an SRAM-based FPGA, these signals mean the configuration-used SRAM’s read/write control signal and their data signals. The abutment of these signals makes development of the programming software easier, because the location of configuration-used SRAM can be described more clearly in the architecture file. The chip-wise connection of global signals is completed through IOC abutment in Area *d*, and Area *e* is reserved for signals in the IOB local routing pool.

**4.2. Array layout and bonding-out plan**

The organization of IOCs in the IO block array is determined by the FPGA architecture and the bond-out schemes for all targeted packages. The total number and the type assignment of the IOCs need to match all targeted packages. The organization starts with the largest package and reduces the bond-out IOCs while coming down in package size. An ideal IOB array organization aims at even distribution of the bonding wires for all targeted packages. This includes an even distribution of the power pins in each bank of every package. We have used software to assist the IOC bond-out planning. Figure 9 shows the layout of an IOB array in an FPGA and its bond-out diagram for the VQ100 and PQ208 packages. The dedicated-function and dual-purpose IOCs are bonded in all targeted packages for programming during the configuration phase. The user IOCs need to be located such that the bond wires are evenly distributed on four sides of the chip. For the smaller packages like VQ100, more user IOCs are skipped in the bonding. The angles of the bonding wires are calculated and compared in the software so as to avoid bonding rule violation and too big a void area in the bonding. The dashed circle in Fig. 9(b) shows a big void inside the bond wires as opposed to the evenly distributed bond wires in Fig. 9(c).

**4.3. Performance**

In Table 2, the input and output propagation delays are

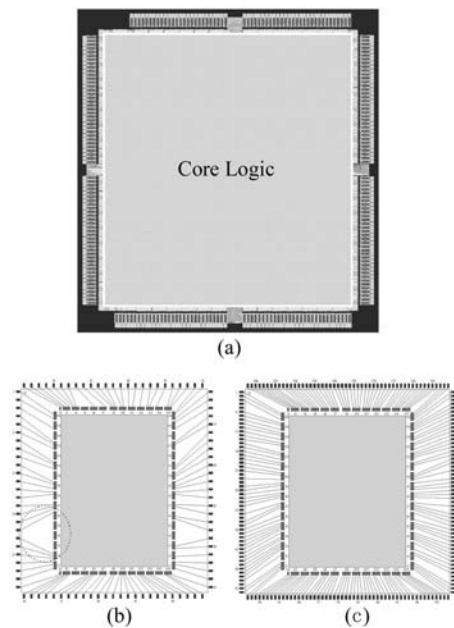


Fig. 9. Layout and package bonding: (a) Layout of IO block array; (b) Bad bonding for VQ100; (c) Bonding for PQ208.

compared between our SPICE simulation data (at worst PVT cases) and two commercial FPGAs, Stratix EP1S20 from Altera and Vertex II Pro XC2VP20 from Xilinx, both based on 0.13 μm CMOS technology.

The comparison of the data is only for reference purposes and is limited to a small subset of the applicable IO standards. The listed values of EP1S20 and XC2VP20 are each from their data sheet<sup>[2,6]</sup>. The “O input to Pad” propagation delay means the time signals transmit from o1/o2 to Pad, and “Pad to I output” delay means the time from Pad to I (o1/o2, Pad, I are shown in Fig. 3). For the listed LVCMOS standards, the IO buffers are tested under the condition of their maximum available drive strength. The table indicates that the propagation delays of our IO buffers deviate from the XC2VP20 data by less than 12%. However, our clock-to-output delays are generally larger than those from XC2VP20. We attribute this to the fact that the configurable registers in our data path are shared by the user data IO and the boundary-scan logic. Thus an additional multiplexer is used for routing and adds an extra delay to the clock-to-output path. Overall, the comparison indicates that our IO cell designs are comparable with these two commercial FPGAs in performance. Better performance can be achieved by more optimized design of the data paths.

**5. Conclusions**

An IO block array in an FPGA is proposed. We present the organization of the array, a novel signal path structure which uses shared registers, the local routing pool for the IOB, and auxiliary power for configurable IO buffers. We also provide a way to generate an abutting layout and multi-package bonding-out. Even though further research is ongoing, preliminary performance comparison of the IO buffer with two commercial FPGAs shows that our design is on the right track. Since the design methodology used in this IOB array is not

Table 2. Comparison with commercial FPGAs.

|  |           | Stratix EP1s20 (ns) | Vertex II Pro XC2VP20 (ns) | Our design (ns) |
|--|-----------|---------------------|----------------------------|-----------------|
| Input propagation delays (Pad to I output) | LVC MOS15 | –                   | 1.36                       | 1.20            |
|  | LVC MOS18 | –                   | 1.27                       | 1.10            |
|  | LVC MOS25 | –                   | 0.91                       | 0.90            |
|  | LVC MOS33 | –                   | 0.96                       | 0.90            |
|  | LVDS25    | –                   | 1.31                       | 1.20            |
| Output propagation delays (O input to Pad) | LVC MOS15 | –                   | 2.90                       | 2.90            |
|  | LVC MOS18 | –                   | 2.68                       | 2.70            |
|  | LVC MOS25 | –                   | 2.35                       | 2.30            |
|  | LVC MOS33 | –                   | 2.46                       | 2.40            |
|  | LVDS25    | –                   | 2.56                       | 2.50            |
| Clock-to-output delays                     | LVC MOS15 | 4.00                | 2.98                       | 3.40            |
|  | LVC MOS18 | 3.43                | 2.74                       | 3.30            |
|  | LVC MOS25 | 2.76                | 2.43                       | 2.90            |
|  | LVC MOS33 | 2.60                | 2.54                       | 3.00            |
|  | LVDS25    | 2.39                | 2.64                       | 3.10            |

limited to any special FPGA architecture, we believe our IOB array design method has a place for meaningful extension in the area of application-specific FPGAs.

## References

- [1] Xilinx Inc. Spartan-3 FPGA family: complete data sheet, 2005
- [2] Xilinx Inc. Virtex-II pro platform FPGAs: complete data sheet, 2003
- [3] Betz V, Rose J, Marquardt A. Architecture and CAD for deep-submicron FPGAs. Boston: Kluwer Academic Publishers, 1999
- [4] Lemieux G, Lewis D. Design of interconnection networks for programmable logic. Boston: Kluwer Academic Publishers, 2004
- [5] Kuon I C. Automated FPGA design, verification and layout. Master Thesis, University of Toronto, 2004
- [6] Altera Corporation. Stratix device family data sheet, Version 3.2, 2005
- [7] Ma Xiaojun, Tong Jiarong. Boundary-scan test circuit designed for FPGA. 5th IEEE International Conference on ASIC Proceedings, 2003, 2: 1190
- [8] Tyhach J, Wang B, Sung C. A 90 nm FPGA IO buffer design with 1.6 Gbps data rate for source-synchronous system and 300 MHz clock rate for external memory interface. IEEE J Solid-State Circuits, 2005, 40(9) :1829
- [9] Dabral S, Maloney T J. Basic ESD and I/O design. Chapter 5. New York: John Wiley & Sons Inc, 1998: 202
- [10] Ni Minghao. Research on automated design methodology of FPGA—the generation and verification of FPGA behavioral model. Doctor Thesis, Institute of Semiconductors, CAS, 2008 (in Chinese)
- [11] Zhou Huabing. Research on FPGA architectures and realization, optimization & verification of FPGA design tool software. Doctor Thesis, Institute of Semiconductors, CAS, 2008 (in Chinese)