

# INTERNATIONAL JOURNAL FOR ADVANCE RESEARCH IN ENGINEERING AND TECHNOLOGY

WINGS TO YOUR THOUGHTS.....

## Design and Implementation of SDRAM Controller in FPGAs

K.Thilagavathi<sup>1</sup>, D.Naramdha<sup>2</sup>, K.Gayathri<sup>3</sup>, F.Sahul Hameed<sup>4</sup>, A.Md.Mian<sup>5</sup>

PG Scholar<sup>1,2,3</sup>, Assistant Professor<sup>4</sup>, Associate Professor<sup>5</sup>  
C.Abdul Hakeem College of Engineering and Technology,  
Melvisharam-632509

skthilaga03@gmail.com<sup>1</sup>, srinana1822@gmail.com<sup>2</sup>, kgayathri.be.ece@gmail.com<sup>3</sup>, bfjsahul@gmail.com<sup>4</sup>,  
ammian79@gmail.com<sup>5</sup>

**Abstract--** Memory performance has become the major bottleneck to improve the overall performance of the computer system. SDRAM controller design, located between the SDRAM and the bus master which reduces the user's effort to deal with the SDRAM command interface by providing a simple generic system interface to the bus master. Opening and closing banks costs memory bandwidth, so we propose the SDRAM Controller design to monitor and manage the status of the four banks simultaneously. This enables the controller to intelligently open and close banks only when necessary. For clock changes and efficient data transfer we introduce wishbone bus master. Several techniques and methods, i.e. data and command buffers based on asynchronous FIFO, multiple bus width adaptation, and improved WISHBONE bus interface are proposed in this paper for fulfilling the requirements. The controller core has been implemented using Xilinx ISE Implementation tool; results show that the memory controller of our design is proven to be efficient, flexible, Programmable, reusable in general purpose FPGAs.

**Keywords:** Command buffer, FIFO, Memory controller, SDRAM, Wishbone bus master.

### 1. INTRODUCTION

With the improvement of the semiconductor process level, microprocessor performance has improved at a rate of 60% per year, while the DRAM latency and bandwidth improve slowly, at only 7%-15%. Because of the growing performance gap between processor and memory, memory has become a major bottleneck to further improve the performance of computer systems [11]. In order to improve the memory bandwidth, reduce memory access latency, a modern memory device allows pipelined access, support bank and open-page technology [2], [6] such as DDR SDRAM Series memory devices, RLD RAM memory [5]. Synchronous DRAMs have become the memory standard in many designs. They provide substantial advances in DRAM performance. They synchronously burst data at clock speeds presently up to 143MHz. They also provide hidden precharge time and the ability to randomly change column addresses on each clock cycle during a burst cycle. There are several limits on DRAM performance. Most noted is the read cycle time, the time between successive read operations to an open row. This time decreased from 10 ns for 100 MHz SDRAM to 5 ns for DDR-400, but has remained relatively unchanged through DDR2-800 and DDR3-1600 generations. However, by operating the interface circuitry at increasingly higher multiples of the fundamental read rate, the achievable bandwidth has increased rapidly.

According to a high-performance DRAM study on earlier versions of DRAM, DRAM's refresh cycle is one reason DRAM is slower than SRAM [2]. The cells of DRAM use sense amplifiers to transmit data to the output buffer in the case of a read and transmit data back to the memory cell in the case of a refresh. During a refresh cycle, the sense amplifier reads the degraded value on a capacitor into a D-Latch and writes back the same value to the capacitor so it is charged correctly for 1 or 0 [7]. Since all rows of memory must be refreshed and the sense amplifier must determine the value of a, already small, degenerated capacitance, refresh takes a significant amount of time. The refresh cycle typically occurs about every 64 milliseconds. The refresh rate of the latest DRAM (DDR3) is about 1 microsecond. In order to decrease latency, SDRAM utilizes a memory bus clock to synchronize signals to and from the system and memory. Synchronization ensures that the memory controller does not need to follow strict timing; it simplifies the implemented logic and reduces memory access latency. With a synchronous bus, data is available at each clock cycle. SDRAM divides memory into two to four banks for concurrent access to different parts of memory. Simultaneous access allows continuous data flow by ensuring there will always be a memory bank read for access. The addition of banks adds another segment to the addressing, resulting in a bank, row, and column address. The memory controller determines if an

# INTERNATIONAL JOURNAL FOR ADVANCE RESEARCH IN ENGINEERING AND TECHNOLOGY

*WINGS TO YOUR THOUGHTS.....*

access addresses the same bank and row as the previous access, so only a column address strobe must be sent. This allows the access to occur much more quickly and can decrease overall latency.

## 2. PROBLEM DESCRIPTION AND OUR BASIC IDEA

### 2.1 Structure

The DRAM is called dynamic because the value of each bit is represented as a charge of a small capacitor, which discharges due to leakage over time. The capacitors must be refreshed periodically to preserve the valid value. In addition to the capacitor, the bit cell contains a pass transistor, which is enabled when the value is read or written. The bits are not accessible individually; instead, they are organized in arrays of rows columns. A row must be prepared before a bit of relevant column can be read. This requires two steps: (1) precharge – the bit-lines (columns) are charged to midpoint voltage between logical 0 and 1; (2) activate – the capacitors of the single row are connected to the bit-lines. During the activation the charge of the capacitor creates a small voltage swing on the bit-line, which is recovered to full voltage by the sense amplifier. Both steps contribute significantly to the latency of the DRAM access, because the bit-line runs over the thousands of rows and has huge capacitance. However,, once the row has been activated its columns can be read/written with lower latency.

To increase the throughput, a memory device contains several independent arrays called banks. The data can be transferred from one bank while other banks are precharging or activating. Similarly, few devices can be connected to the same SDRAM interface and enabled by chip-select. This address dimension is called rank.

### 2.2 Operation

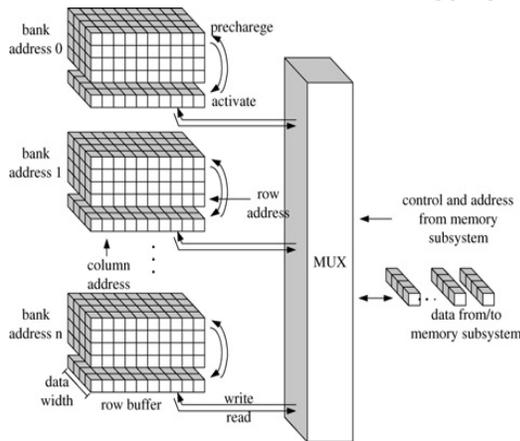
SDRAMs have a three dimensional layout. The three dimensions are banks, rows and columns. A bank stores a number of word-sized elements in rows and columns, as shown in Figure 1. On an SDRAM access, the address of the request is decoded into bank, row and column addresses using a memory map. A bank has two states, idle and active. The bank is activated from the idle state by an *activate* (ACT) command that loads the requested row onto a row buffer, which stores the most recently activated row. Once the bank has been activated, column accesses such as *read* (RD) and *write* (WR) bursts can be issued to access the columns in the row buffer. These bursts have a programmable burst length (BL) of 4 or 8 words (for DDR2 SDRAM). Finally, a Precharge

(PRE) command is issued to return the bank to the idle state. This stores the row in the buffer back into the memory array. Read and write commands can be issued with an auto-precharge flag resulting in an automatic precharge at the earliest possible moment after the transfer is completed. In order to retain data, all rows in the SDRAM have to be refreshed regularly, which is done by precharging all banks and issuing a *refresh* (REF) command. If no other command is required during a clock cycle, a *no-operation* (NOP) command is issued. SDRAM devices are typically divided into four banks. These banks must be opened before a range of addresses can be written to or read from. The row and bank to be opened are registered coincident with the Active command. When a new row on a bank is accessed for a read or a write it may be necessary to first close the bank and then re-open the bank to the new row. Closing a bank is performed using the Precharge command. Opening and closing banks costs memory bandwidth, so the SDRAM Controller Core has been designed to monitor and manage the status of the four banks simultaneously. This enables the controller to intelligently open and close banks only when necessary.

When the Read or Write command is issued, the initial column address is presented to the SDRAM devices. The initial data is presented concurrent with the Write command. For the read command, the initial data appears on the data bus 1-4 clock cycles later. This is known as CAS latency and is due to the time required to physically read the internal DRAM and register the data on the bus. The CAS latency depends on the speed grade of the SDRAM and the frequency of the memory clock. In general, the faster the clock, the more cycles of CAS latency is required. After the initial Read or Write command, sequential read and writes will continue until the burst length is reached or a Burst Terminate command is issued. SDRAM devices support a burst length of up to 8 data cycles. The SDRAM Controller Core is capable of cascading bursts to maximize SDRAM bandwidth. SDRAM devices require periodic refresh operations to maintain the integrity of the stored data. The SDRAM Controller Core automatically issues the Auto Refresh command periodically. No user intervention is required. The Load Mode Register command is used to configure the SDRAM operation.

# INTERNATIONAL JOURNAL FOR ADVANCE RESEARCH IN ENGINEERING AND TECHNOLOGY

WINGS TO YOUR THOUGHTS.....



**Figure 1:** Multi-Banked Architecture Of SDRAM

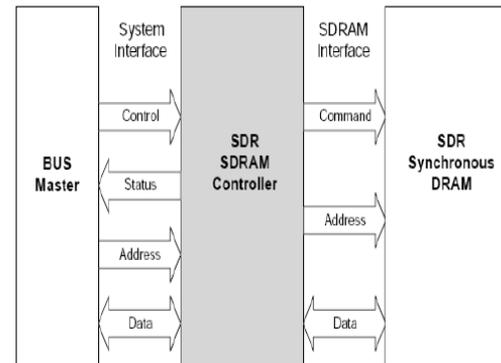
This register stores the CAS latency, burst length, burst type, and write burst mode. The SDR controller only writes to the base mode register. To reduce pin count, SDRAM row and column addresses are multiplexed on the same pins.

### 3. THE DESIGN OF SDRAM CONTROLLER

The functions that SDRAM memory controller needs to be done include: to receive and process the memory access requests, memory self-test operation, memory refresh and initialization operation, implementation of CPU register access path, read and write of all register in SDRAM controller. The process of SDRAM controller for memory access requests is: receiving all kinds of requests and scheduling the memory access according to the priority of the request, generating the correct memory address control signals in accordance with the selected request and finishing the sending and accepting of memory data. This SDRAM controller reference design, located between the SDRAM and the bus master, reduces the user's effort to deal with the SDRAM command interface by providing a simple generic system interface to the bus master. Figure 1 shows the relationship of the controller between the bus master and SDRAM. The bus master can be either a microprocessor or a user's proprietary module interface.

#### 3.1 Overall Structure

The SDRAM memory controller that we designed is comprised of wishbone bus handler, SDRAM controller as shown in Figure 3. Wishbone bus handler take necessary care about domain clock change over. The memory access control module includes four sub-blocks such as SDRAM Bus converter, SDRAM request generator bank controller, SDRAM transfer

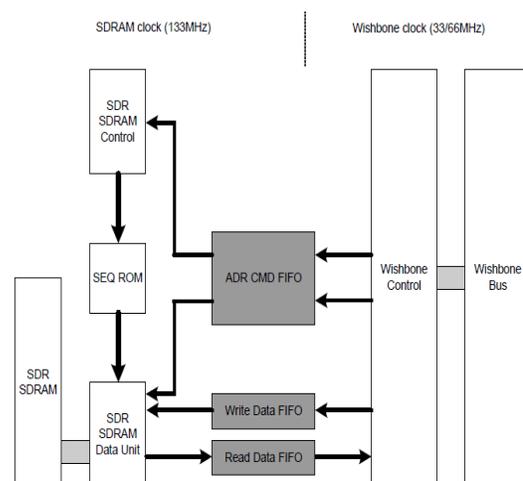


**Figure 2:** Structure of SDRAM Controller Core

controller. SDRAM interface module implements the physical interface of memory, includes transmission of address and command, data transmission and generation of memory clock signal. Because memory control module is the key part of SDRAM controller.

#### 3.2 Functions of SDRAM Controller

SDRAM controller is the most critical part in the memory controller. As the memory chip has 4 banks, in order to solve the problem of address conflicts as well as to facilitate to implement multi-bank concurrent memory access requests are divided into four request groups in accordance with their bank address. Memory controller receives memory read and write request, initialization sequence request from the register control module, refresh request and self-test request. Initialization sequence request only run once at system start up and can generate pre-charge, load mode register and refresh requests.



**Figure 3:** Structure of SDRAM Controller

Upon completion of initialization, automatically refresh request, self-test request initiated by CPU and memory access requests must be sent to the

# INTERNATIONAL JOURNAL FOR ADVANCE RESEARCH IN ENGINEERING AND TECHNOLOGY

*WINGS TO YOUR THOUGHTS.....*

SDRAMN controller, and then SDRAM controller request generation module generate SDRAM controller signals for selected request, including the address control signal, data strobe control signal, and read and write control signal. SDRAM controller signals are then sent to the SDRAM interface to generate physical interface signals. Wishbone bus handler consists of bus master and protocol handshake between bus master and custom SDRAM controller. And also take necessary care about clock changes. There are three FIFO buffer available in wish bone bus handler which handles synchronous as well as asynchronous clock changes. Command FIFO generates the request according with bank address and application layer data length. Write FIFO generates request for write operation and byte mask. Read FIFO checks the length of the read request. It decides whether the request is valid or not. It also generates the read request accordance with bank address. All requests from wishbone bus master are bidirectional.

SDRAM bus converter converts and re-aligns the system side 32 bit into equivalent 8/16/32 SDR format. It changes the bus width according with bank transfer control signal. During write transaction, it splits the 32 Bit Application data into 8/16/32 SDRAM Bus width format. During Read transaction, it re-packs the 8/16/32bit SDRAM data into 32 bit Application data. SDRAM request generator does following functions,

- 1) Based on the SDRAM bus width, internal address and burst length will be re modified.
- 2) If the wrap = 0 and current application burst length is crossing the page boundary, then request will be split into two with corresponding change in request address and request length.
- 3) If the wrap = 0 and current burst length is not crossing the page boundary, then request from application layer will be transparently passed on the bank control block.
- 4) If the wrap = 1, then this block will not modify the request address and length. The wrapping functionality will be handling by the bank control module and column address will rewind back as follows XX -> FF....
- 5) Based on column configuration bit, this block generate the column address, row address and bank address.

Note: With Wrap = 0, each request from Application layer will be splits into two request, if the current burst cross the page boundary.

SDRAM bank controller takes requests from SDRAM request generator, checks for page hit/miss and an issue precharge/activate commands and then passes the request to SDRAM Transfer Controller.

SDRAM Transfer Controller takes requests from SDRAM Bank controller, runs the transfer and controls data flow to/from the application layer. At the end of the transfer it issues a burst terminate if not at the end of a burst and another command to this bank is not available.

### 3.3 SDRAM Interface

Prior to normal operation, SDRAM must be initialized. The following sections provide detailed information covering device initialization, register definition, command descriptions and device operation. SDRAM must be powered up and initialized in a predefined manner. Operational procedures other than those specified may result in undefined operation. Once power is applied to VDD and the clock is stable, the SDRAM requires a 100µs delay prior to issuing any command other than a COMMAND INHIBIT or NOP. Starting at some point during this 100µs period and continuing at least through the end of this period, COMMAND INHIBIT or NOP commands should be applied.

Once the 100µs delay has been satisfied with at least one COMMAND INHIBIT or NOP command having been applied, a PRECHARGE command should be applied. All device banks must then be precharge, thereby placing the device in the all banks idle state. Once in the idle state, two AUTO REFRESH cycles must be performed. After two refresh cycles are complete, SDRAM ready for mode register programming. Because the mode registers will power up in unknown state, it should be loaded prior to applying any operational command.

### 3.4 Mode Register

The mode register is used to define the specific mode of operation of SDRAM. This definition includes the selection of burst length, a burst type, CAS latency, an operating mode and a write burst mode as shown in the Mode Register Definition Diagram below. The mode register is programmed via the LOAD MODE REGISTER command and will retain the stored information until it is programmed again or the device loses power.

Mode register bits M0-M2 specify the burst length, M3 specifies the type of burst (sequential or interleaved), M4-M6 specify the CAS latency, M7-M8 specify the Operating mode, M9 specifies the write burst mode, M10 and M11 are reserved for future use. M12 is undefined but should be driven LOW during loading of the mode register. The mode register must be loaded when all device banks are idle, and the Controller must wait the specified time before initiating the subsequent operation.

# INTERNATIONAL JOURNAL FOR ADVANCE RESEARCH IN ENGINEERING AND TECHNOLOGY

*WINGS TO YOUR THOUGHTS.....*

### 3.5 Operating Mode

The normal operating mode is selected by setting M7 and M8 to zero; the others combinations of values for M7 and M8 are reserved for future use and/or test modes. The programmable burst length applies to both READ and WRITE bursts.

### 3.6 Writing Burst Mode

When M9 = 0, the burst length programmed via M0-M2 applies to both READ and WRITE bursts; when M9 = 1, the programmed burst length applies to READ bursts, but write accesses are single-location (non-burst) accesses.

### 3.7 Burst Length

The burst length determines the maximum number of column locations that can be accessed for a given READ or WRITE command. Burst lengths of 1, 2, 4 or 8 locations are available for both the sequential and the interleaved burst types, and full-page burst is available for sequential type only. The full-page burst is used in conjunction with the BURST TERMINATE command to generate arbitrary burst lengths. When a READ or WRITE command is issuing, a block of columns equal to burst length is effectively selected. All accesses for that burst take place within this block, meaning that the burst will wrap within the block if a boundary is reached.

### 3.8 Burst Type

The sequential or interleaved burst is selected via bit M3. The ordering of accesses within a burst is determined by the burst length, the burst type and starting column address.

## 4. IMPLEMENTATION OF SDRAM CONTROLLER

### 4.1 Simulation

SDRAM Controller design consists of two major module i.e. Wishbone bus handler and a SDRAM control module. Simulation of SDRAM controller has been done using Model simulator. Modelsim is a widely used simulation tool to simulate and debug the digital circuits. In order to fully verify this SDRAM controller implementation, a large simulation environment for functional verification has been built. This environment includes the SDRAM controller RTL model, as many as different kinds of models such as Spartan 3E, Spartan 3A, Vertex series which are provided by Xilinx. In this controller design improved wishbone bus interfaced used with registered feedback cycle to implement advanced synchronous burst termination.

We analyze the performance improvement of this on-chip bus interface by taking the SDRAM access as an example. Two times write only take two cycles in the view of the master side, while classical interface has to take 4 cycles. Similarly, if we define the read operation time is the interval time between the first request sending out and the last data arriving, then the improved interface proposed in this paper only takes 5 cycles, while the classical takes 8 cycles. The speedup is growing with the number of launched outstanding access requests, which is shown in Table 1. Another important advantage of this improved interface is that it cuts off the combinational logic feedback path, which starts from the master, runs through the slave and finally feedback the master. It brings a great benefit to the timing closure in synthesis phase, which can achieve higher operation frequency. And the improved interface provides an excellent solution for clock domains crossing transfer which is impossible to be solved by the classical.

**Table 1:** Performance Comparison Of Improved Bus Interface

No of write/read	CLASSICAL (cycles)	IMPROVED (cycles)	SPEEDUP
2 Writes	4	2	50%
4 Writes	8	4	50%
2 Reads	8	5	37.5%
4 Reads	16	7	56.25%
8 Reads	32	11	65.625%

### 4.2 Analysis of SDRAM controller

The FPGA prototype verification has also been done with 82.682 MHz SDRAM Clock and Wishbone clock of 116.973MHz clock frequency. Xilinx ISE implementation tool is used for analysis and implementation. Reports shows that the SDRAM controller architecture proposed in this paper can operate stably at 143 MHz, XC3S700A Spartan 3A/3AN series Xilinx FPGA board used for this project. They are organised as 8/16/32 bit words in 4 banks of 4096 rows by 256 columns. The chip is speed grade -4 so, according to the specification; they can be clocked upto 143MHz (with CAS latency (CL) of 3) or 100MHz (with CAS latency (CL) of 2). The maximum combinational path delay is 9.678ns. The utilization of device for this controller design is explained in synthesis results which are shown in table 2.

# INTERNATIONAL JOURNAL FOR ADVANCE RESEARCH IN ENGINEERING AND TECHNOLOGY

WINGS TO YOUR THOUGHTS.....

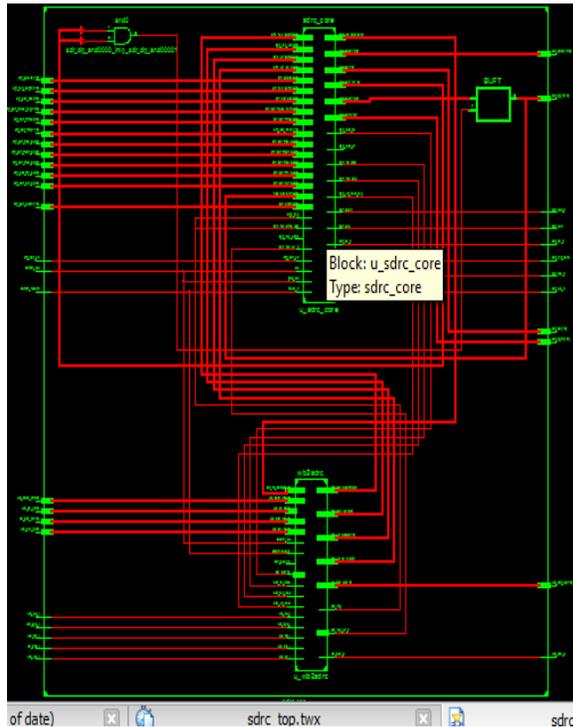


Figure 4: RTL View of SDRAM Controller

Table 2:  
Device Utilization Summary

Logic Utilization	Used	Available	Utilization
Number of Slice Flip Flops	584	11,776	4%
Number of 4 input LUTs	1,275	11,776	10%
Number of occupied Slices	812	5,888	13%
Number of Slices containing only related logic	812	812	100%
Number of Slices containing unrelated logic	0	812	0%
Total Number of 4 input LUTs	1,329	11,776	11%
Number used as logic	1,065		
Number used as a route-thru	54		
Number used for Dual Port RAMs	194		
Number used as Shift registers	16		
Number of bonded IOBs	195	372	52%
Number of BUFGMUXs	2	24	8%
Average Fan-out of Non-Clock Nets	3.47		

## 5. CONCLUSION

SDRAM memory supports the multi-bank architecture. It has higher performance compared to DRAM Memory Controller. In order to fully exhibit

the advantages of SDRAM performance, efficient design of memory controller is essential. In this paper we presented an SDRAM controller that provides the constant and well-known access time to the SDRAM memory. All of the new techniques can co-operate very well and make the whole architecture achieve high throughput and high compatibility. We have studied the SDRAM specifications in-depth, under the premise of meeting memory timing constrain; a memory controller based on SDRAM is designed. The synthesis results and RTL view clearly shows the timing constrains. In future, this entire controller core will be implemented in Xilinx Memory Expansion Module.

## References

- [1] Pan Guoteng, Luo Li, Ou Guodong, Dou Qiang, Xie Lunguo, "Design and Implementation of DDR3 Controller". Third International Conference on Intelligent System Design and Engineering Applications, 2013.
- [2] Dimitris Kaseridis, Jeffrey Stuecheli, Lizy Kurian John. Minimalist "Open-Page: a DRAM page-mode scheduling policy for the manycore" era. Proceedings of the 44th Annual IEEE/ACM International Symposium on Micro architecture. pp. 24-35, 2011.
- [3] Yuan Xie. "Modelling, Architecture, and Applications for Emerging Memory Technologies". IEEE Design & Test. 28(1):44-51, 2011.
- [4] Woo young Jang, Student Member, IEEE, and David Z. Pan, Senior Member, IEEE, "Application-Aware NOC Design for Efficient SDRAM Access". IEEE Transactions on Computer-Aided Design of Integrated Circuits And Systems, Vol. 30, No. 10, October 2011.
- [5] Micron RLDRAM memory: "Unparalleled Bandwidth and Low Latency" [EB/OL].<http://www.micron.com/products/Dram modules/lrdimm> [2012-08-20].
- [6] Grannaes, M., Jahre, M., Natvig, L. "Low-Cost Open-Page Prefetch Scheduling in Chip Multiprocessors". IEEE International Conference on Computer Design (ICCD 2008), pp390-396, 2008
- [7] Benny Akesson, Kees Goossens, Markus Ringhofer, "a Predictable SDRAM Memory Controller", Netherlands, Austria, 2007.
- [8] Dong Wang, J. Ma, S. Chen, Y. Guo, "The Design and Analysis of a High Performance Embedded External Memory Interface",

# INTERNATIONAL JOURNAL FOR ADVANCE RESEARCH IN ENGINEERING AND TECHNOLOGY

*WINGS TO YOUR THOUGHTS.....*

Proceedings of the Second International Conference on Embedded Software and Systems, 2005.

- [9] Specification for the WISHBONE System-on-Chip (SoC) Interconnection Architecture for Portable IP Cores. Rev.B3, Open cores Organization, September 7. 2002.
- [10] Clifford E. Cummings, Peter Alfke, "Simulation and Synthesis Techniques for Asynchronous FIFO Design with Asynchronous Pointer Comparisons", SNUG-2002, San Jose, CA, 2002.
- [11] Bruce Jacob, Spencer Ng, David Wang. Memory Systems: "Cache, DRAM, Disk". Morgan Kaufmann Publishers Inc., San Francisco, CA, 2007.