FPGA Design for Implementing data acquisition using SDRAM

Ankit Jain[1], Dr. Uday Bhanu Singh Chandrawat[2]*

[1] Department of Electronics and Communication Engineering,
Indore, M.P., India
ankit96927@gmail.com

[2] Professor & Head, Department of Electronics and Communication Engineering,*
Swami Vivekanand College of Engineering., Khandwa Road, Indore, M.P., India

Abstract:-

Design and implementation of a Data Acquisition System (DAS) in Field Programmable Gate Arrays (FPGA).Data is coming with a very fast rate from a single channel pipeline 12-bit, 500-MSPS analog to-Digital converter (ADS5463), which provides LVDS compatible output, module will accept data From ADC on every positive edge of the clock cycle and generate the control signal to highly Optimized CORESDR (Soft IP module), CORESDR generate all the signal required for external SDRAM module like CS#, RAS#, CAS#, WE# CKE and other control signal require for SDRAM. This project provides automatic address incrementing facility and configuration register, that can be Configured at a run time for different speed grade, write and read burst size. In this project, we are Using three such main module and main module controller on a single FPGA chip so that one can Able to transfer data into SDRAM with a very fast rate more than 100 mhz for infinitely long time With 100% reliability without loss of single word, and read data with a slower speed and transfer data Into DSP for further analysis. A tool used for this project "Actel Libero project manager" for programming, synthesizing, Place and route and timing analysis. It smart design takes care about whole programming flow.

FPGA used: ***proasic3 A3P400.***

ACTEL LIBERO FPGA TOOL

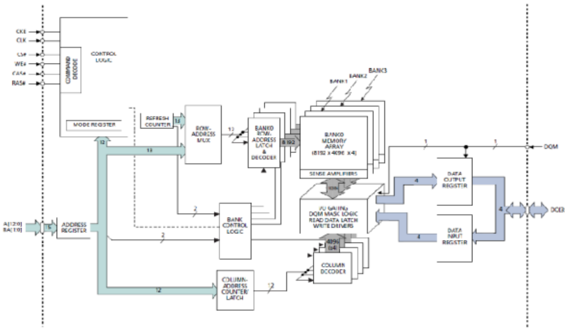
Actel's Libero IDE offers the latest and best-in-class tools from leading EDA vendors such as Mentor Graphics, Synapti CAD, and Synplicity. These tools, combined with custom-developed tools from Actel, are all integrated into a single FPGA development package. The Libero IDE flow includes a powerful Project Manager that guides you through the design process, keeps track of your design files, and manages file exchanges between the various tools. Libero IDE includes Actel's Designer software, which offers premier backend tools for physical implementation. Designer is available as a standalone product, for those who want to use their own design and verification tools.

Description

The 512Mb SDRAM is a high-speed CMOS, dynamic random-access memory containing 536,870,912 bits. It is internally configured as a quad-bank DRAM with a synchronous interface (all signals are registered on the positive edge of the clock signal, CLK). Each of the x4's 134,217,728 bit banks is organized as 8192 rows by 4096 columns by 4 bits. Each of the x8's 134,217,728-bit banks is organized as 8192 rows by 2048 columns by 8 bits. Each of the x16's 134,217,728-bit banks is organized as 8192 rows by 1024 columns by 16 bits. Read and write accesses to the SDRAM are burst-oriented; accesses start at a selected location and continue for a programmed number of locations in a programmed sequence. Accesses begin with the registration of an ACTIVE command, which is then followed by a READ or WRITE command. The address bits registered coincident with the ACTIVE command are used to select the bank and row to be accessed (BA[1:0] select the bank; A[12:0] select the row). The address bits registered coincident with the READ or WRITE command are used to select the starting column location for the burst access. The SDRAM provides for programmable read or write burst lengths (BL) of 1, 2, 4, or 8 locations, or the full page, with a burst terminate option. An auto precharge function may be enabled to provide a self-timed row precharge that is initiated at the end of the burst sequence. The 512Mb SDRAM uses an internal pipelined architecture to achieve high-speed operation. This architecture is compatible with the 2*n* rule of prefetch architectures, but it also allows the column address to be changed on every clock cycle to achieve a high speed, fully random access. Precharging one bank while accessing one of the other three banks will hide the PRECHARGE cycles and provide seamless, high-speed, random access operation. The 512Mb SDRAM is designed to operate in 3.3V memory systems. An auto refresh mode is provided, along with a power-saving, power-down mode. All inputs and outputs are LVTTL compatible. SDRAMs offer substantial advances in DRAM operating performance, including the ability to synchronously burst data at a high data rate with automatic column-address generation, the ability to interleave between internal banks to hide precharge time, and the capability to randomly change column addresses on each clock cycle during a burst access.

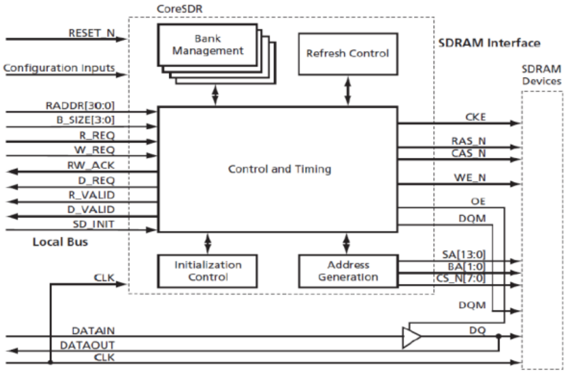
Control Signal Interpretation

Function	Cs*	Ras*	Cas*	We*
Command Inhibit	H	X	X	X
Nop	H	X	X	X
Load Mode Register	L	L	L	L
Auto /Self Refresh	L	L	L	H
Precharge	L	L	H	L
Active	L	L	H	H
Write	L	H	L	L
Read	L	H	L	H
Burst Terminate	L	H	H	L



CORESDR IP

CoreSDR provides a high-performance interface to single-data-rate (SDR) synchronous dynamic random access memory (SDRAM) devices. CoreSDR accepts read and write commands using the simple local bus interface and translates these requests to the command sequences required by SDRAM devices. CoreSDR also performs all initialization and refresh functions. CoreSDR uses bank management techniques to monitor the status of each SDRAM bank. Banks are only opened or closed when necessary, minimizing access delays. Up to four banks can be managed at one time. Access cascading is also supported; allowing read or writes requests to be chained together. This results in no delay between requests, enabling up to 100% memory throughput for sequential accesses. CoreSDR is provided with configurable memory settings (row bits and column bits) and timing parameters (CAS latency, tRAS, tRC, tRFC,tRCD, tRP, tMRD, tRRD, tREFC, tWR). The memory settings are configured through the Smart Design GUI.



SDRAM Controller Parameters

Parameter	Port Bits	Valid Values	Description
RAS	4	1–10	SDRAM active to precharge (tRAS), specified in clock cycles
RCD	3	2–5	SDRAM active to read or write delay (tRCD), specified in clock
RRD	2	2–3	SDRAM active bank *a* to active bank *b* (tRRD), specified in clock
RP	3	1–4	SDRAM *precharge* command period (tRP), specified in clock
RC	4	3–12	SDRAM *active* to *active /auto-refresh* command period (tRC), specified in clock cycles
RFC	4	2–14	*Auto-refresh* to *active/auto-refresh* command period (tRFC), specified in clock cycles
MRD	3	1–7	SDRAM *load mode register* command to *active* or *refresh* command (tMRD), specified in clock cycles
CL	3	1–4	SDRAM CAS latency, specified in clock cycles
BL	2	0–3	SDRAM maximum burst length (encoded). Values are decoded as follows: 0: 1 transfer/burst 1: 2 transfers/burst 2: 4 transfers/burst 3: 8 transfers/burst (valid for SDR only)
WR	2	1–3	SDRAM write recovery time (tWR)
DELAY	16	10–65,535 ns	Delay after a reset event that the controller waits before initializing the SDRAM, specified in clock cycles. Per JEDEC standards, SDR devices require this delay to be a minimum of 200 µs.
REF	16	10–65,535 ns	Period between auto-refresh commands issued by the controller, specified in clock cycles. REF = auto refresh interval / tCK, where tCK is the clock cycle in ns.
COLBITS	3	3–7	Number of bits in the column address (encoded). Values are decoded as follows: 3: 8 column bits - 4: 9 column bits 5: 10 column bits -6: 11 column bits 7: 12 column bits
ROWBITS	2	0–3	Number of bits in the row address (encoded). Values are decoded as follows: 0: 11 row bits 1: 12 row bits 2: 13 row bits 3: 14 row bits
REGDIMM	1	0–1	Set when using registered/buffered DIMMs. Causes adjustment in local bus interface timing to synchronize with SDRAM command timing delayed by register/buffer on DIMM.

Local Bus signals

Signal	N	I/O	Descripti
CLK	Clock	Input	System clock. All local bus signals are synchronous to this
RESET_N	Reset	Input	System reset
RADDR[30:0]	Memory Address	Input	Local bus address
B_SIZE[3:0]	Burst Size	Input	Local bus burst length. Valid values are 1 through BL, where BL is the programmed burst length.
R_REQ	Read Request	Input	Local bus read request
W_REQ	Write Request	Input	Local bus write request
AUTO_PCH	Auto-Precharge Request	Input	When asserted in conjunction with R_REQ or W_REQ, causes command to be issued as *read with auto-precharge* or *write with auto-precharge*, respectively.
RW_ACK	Read/Write Acknowledge	Output	Acknowledgement of read or write request
D_REQ	Data Request	Output	Requests data on the local bus write data bus (datain) during a write transaction. Asserts one clock cycle prior to when data is required.
W_VALID	Write Data Valid	Output	Frames the active data being written to SDRAM. Mimics D_REQ, except that it is delayed by one clock cycle.This signal is typically not used and is retained for legacy compatibility.
R_VALID	Read Data Valid	Output	Indicates that the data on the local bus read data bus (dataout) is valid during a read cycle.
SD_INIT	Initialization Strobe	Input	Causes CoreSDR to reissue the initialization sequence to SDRAM devices. CoreSDR will always issue the initialization sequence (including the startup delay) after reset, regardless of the SD_INIT state. This signal can be tied LOW if runtime reinitialization is not required.

SDR SDRAM Interface Signals

Signal	N	I/O	Descripti
SA[13:0]	Address Bus	Output	Sampled during the *active*, *precharge*, *read*, and *write* commands. This bus also provides the mode register value during the *load mode register* command.
BA[1:0]	Bank Address	Output	Sampled during *active*, *precharge*, *read*, and *write* commands to determine which bank command is to be applied to.
CS_N[7:0]	Chip Selects	Output	SDRAM chip selects
CKE	Clock Enable	Output	SDRAM clock enable. Held LOW during reset to ensure SDRAM dq and dqs outputs are in the high-impedance
RAS_N	Row Address Strobe	Output	SDRAM command input
CAS_N	Column Address Strobe	Output	SDRAM command input
WE_N	Write Enable	Output	SDRAM command input
DQM	Data Mask	Output	SDRAM data mask asserted by controller during SDRAM initialization and during burst terminate. User may sum with user data mask bits.
OE	Output Enable	Output	Tristate control for DQ data

Figure 2. Top Module Block diagram

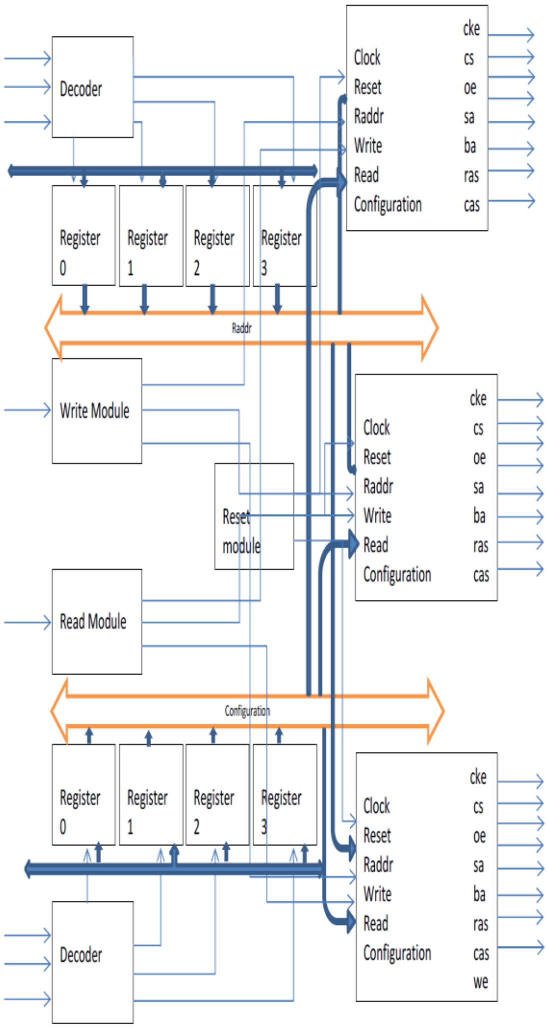


Figure 3. Synthesis Block diagram Top Module

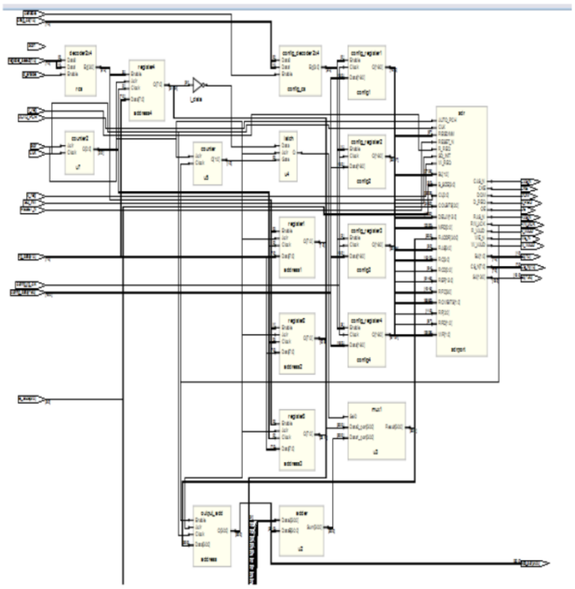


Figure 4. Synthesis Block diagram Main project

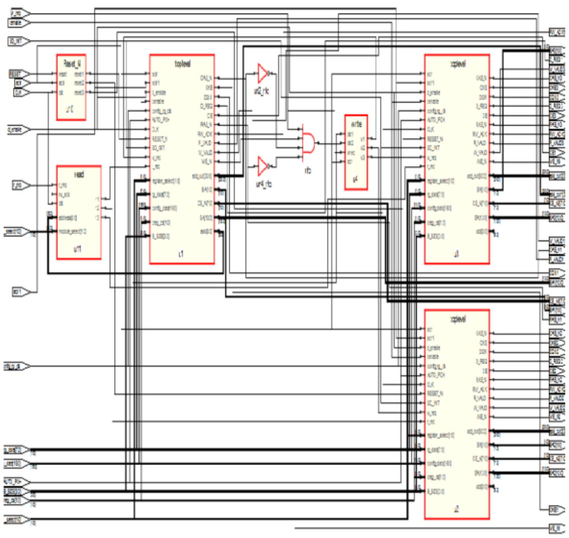
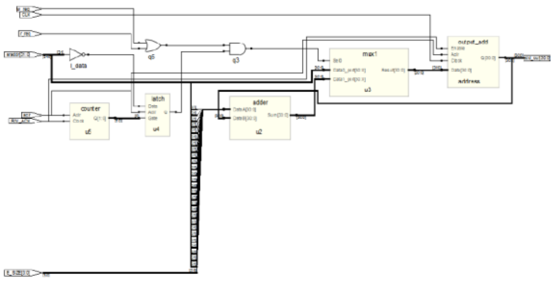


Figure 5. Synthesis Diagram Address Increment Module



TIMING WAVEFORMS

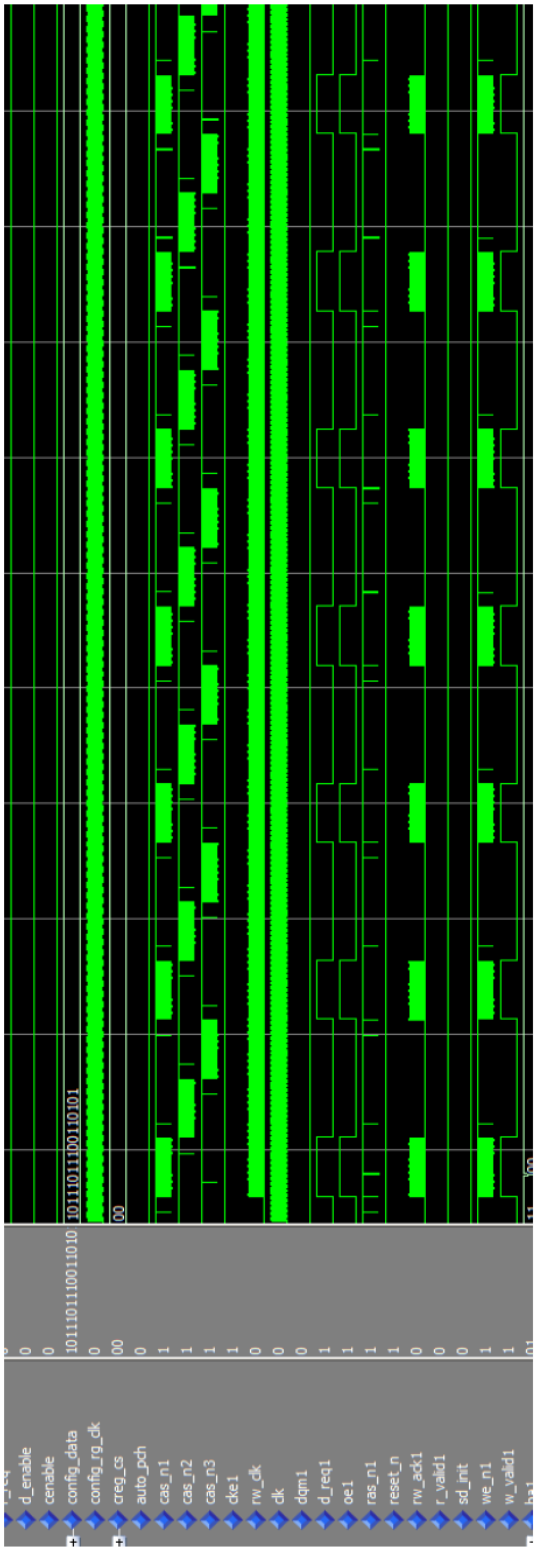


Figure 9. Timing waveform obtained by simulating VHDL code of write module

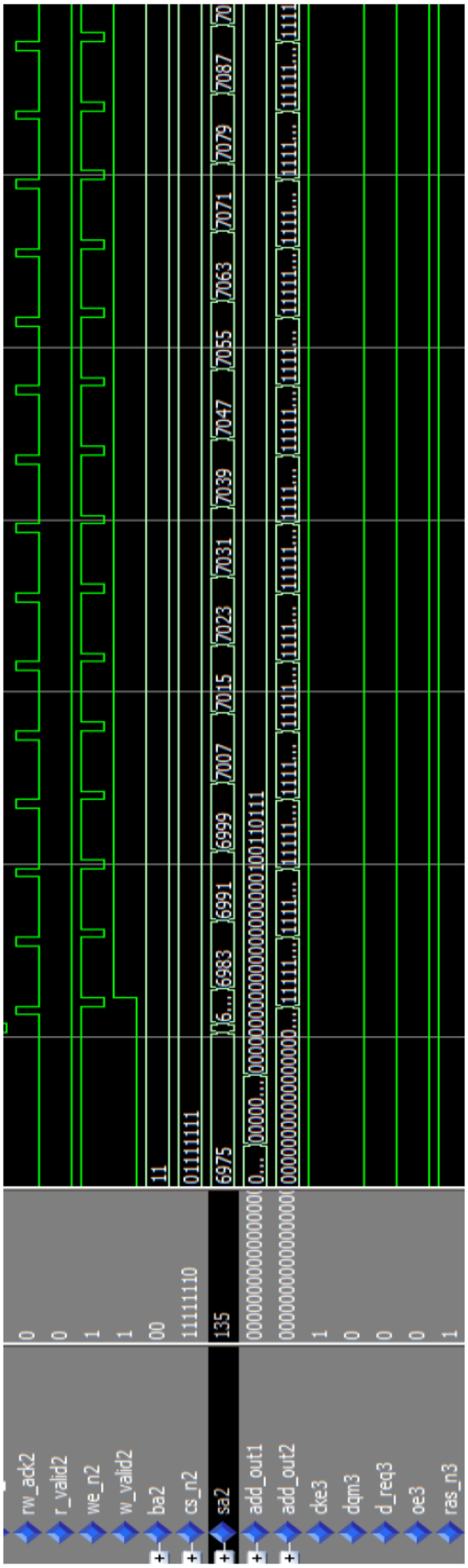


Figure 10. Timing waveform obtained by simulating VHDL code of address increment module

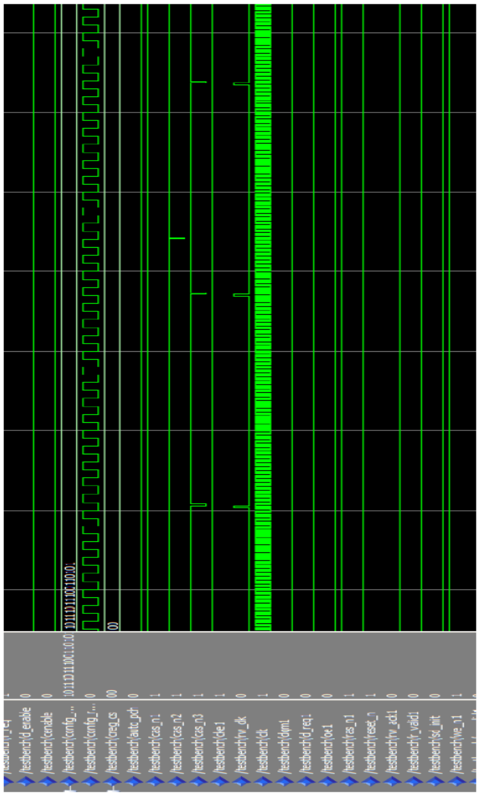


Figure 11. Timing waveform obtained by simulating VHDL code of Read module

REFERENCES

[1] Burchardt, A., Hekstra-Nowacka, E., and Chauhan, A., “A Real-time Streaming Memory Controller”, Design, Automation and Test in Europe 2005 Proceedings, 2005, vol. 3, pp. 20-25.

[2] Heithecker, S. and Ernst, R., “Traffic Shaping for An FPGA Based SDRAM Controller with Complex QoS Requirements”, Design Automation Conference 2005, June 2005, pp. 575-578.

[3]Miloš Drutarovský, Ján Šaliga, Linus Michaeli, Ingrid Hroncová, “Remote Laboratory for FPGA Based Reconfigurable Systems Testing”.

[4] Reza Hashemian, Jason Riddley, “FPGA e-Lab, A Technique to Remote Access a Laboratory to Design and Test”, IEEE Trans. Circuits Syst. II, vol. 49, no. 8, pp. 539–544, Aug. 2002.

[5] W. M. El Medany, “FPGA remote laboratory for hardware e-learning courses”, Computational Technologies in Electrical and Electronics Engineering, Aug 2008, pp106-109.