Implementing a LEON-based System-on-chip

ECE652 Spring'05

Presentation by:

Tushti & Ikram

Introduction

- SoC – major revolution in the IC design
- whole functionality of a system is placed on a single chip
- **Advantages**:
  - High performance
  - Shorter Design Cycle time
  - Helps overcome space constraints
  - is economically viable
- **Challenges:**
  - To deal with deep sub-micron complexities
  - To integrate the IP
  - To meet the mixed-Signal requirements

Goals

- Customize LEON-2 processor
- Make use of AES block from opencores.org
- Create AMBA wrapper to enable communication through AMBA buses.
- Integrate System-on-chip Platform
- Perform physical synthesis

Core selection

- Processor core: LEON2-1.0.12
- Open Cores: AES
- BUS Standard: AMBA AHB/APB
- Artisan RAM – as replacement for Dual Port RAM, SDRAM models.

Features of LEON 2-1.0.12

- It is a 32 bit processor
- Can be configured for technologies – TSMC25, Virtex2, Atmel-18/25/35 etc.
- Debug support unit
- Cache System (Harvard Architecture)
- Other peripherals for embedded applications – UARTS, 24-bit Timers, I/O ports, Memory controller.

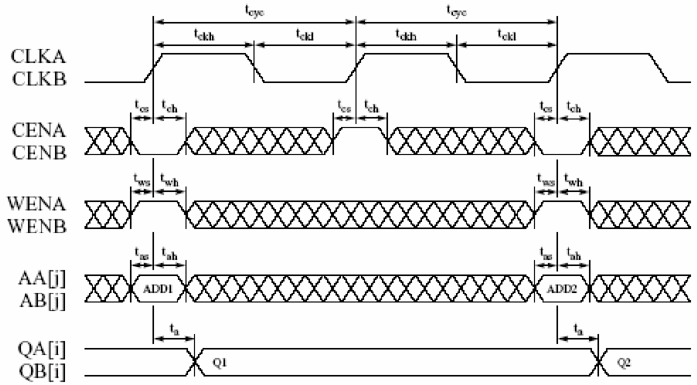
Customizing LEON-2

- Target technology – TSMC 25
- Artisan Rams used as Dual-Port RAM & SDRAM replacement.
- Replaced RAM Behavioral models with corresponding RTL models.
- Boot from External memory

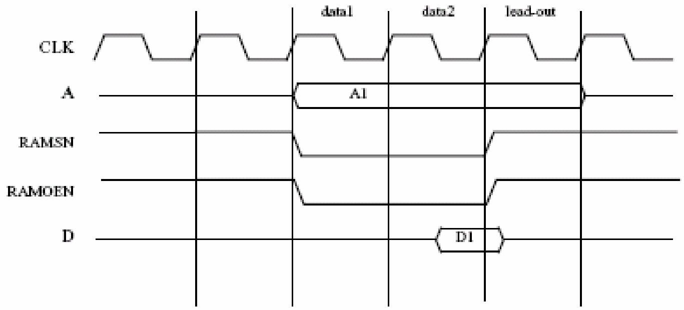
RAM wrappers (*box0.vhd)

- SRAM read operation – address of the memory location to be accessed should already be there when rising edge of clock appears.
- LEON-2 processor read operation – Loads the address at the rising edge of the clock
- Wrapper acts as an interface between Leon-2 and SRAM

Reason for using Wrappers



Rising signals are measured at 50% VDD and falling signals are measured at 50% VDD.



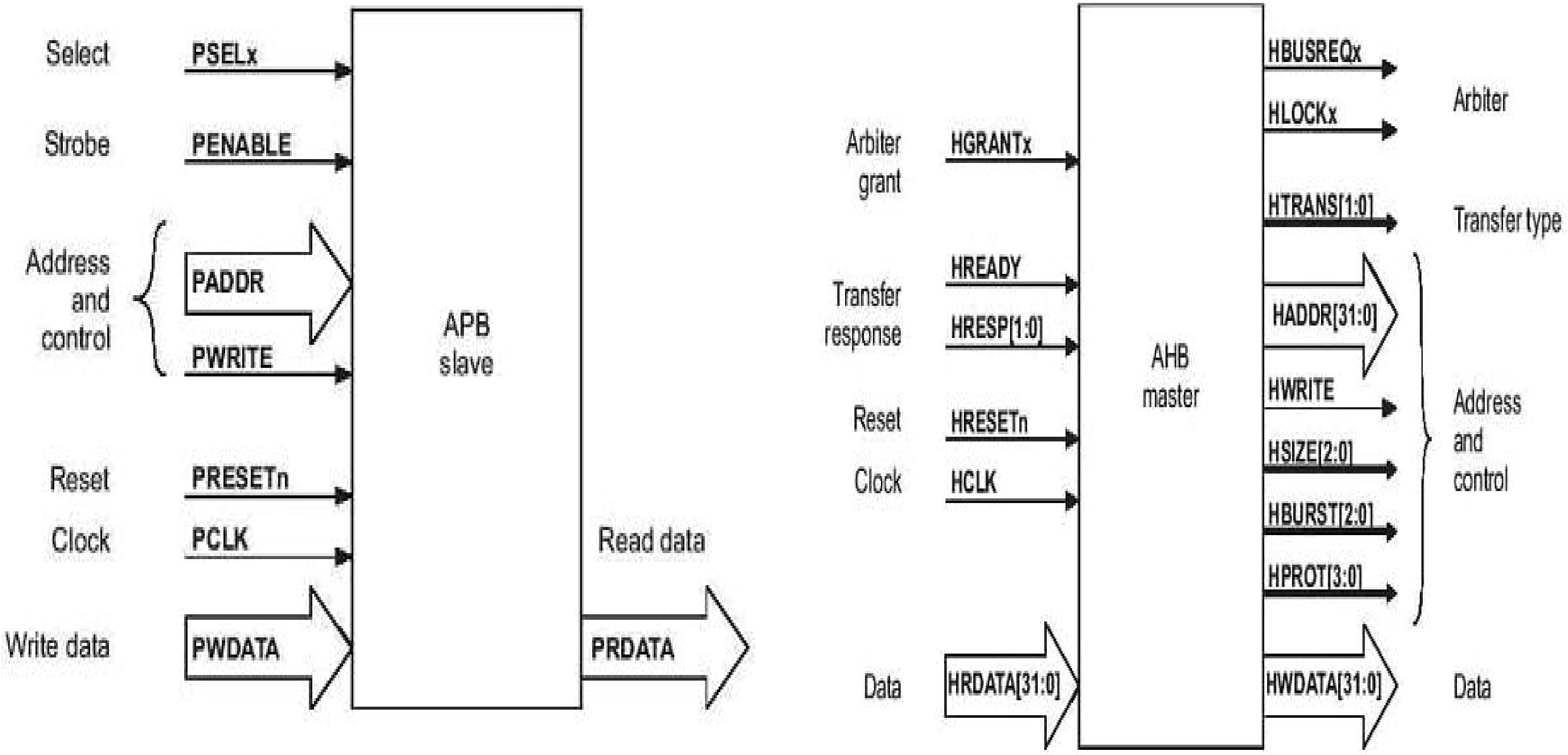
Artisan RAM read cycle

LEON-2 processor read cycle

Design IP blocks (e.g. AES)

- Each IP block need to be put inside a wrapper
- AES act as
    - AMBA AHB Bus Master
    - AMBA APB Bus Slave
- Next slide shows the interface of a standard AHB Bus Master and APB Bus slave.
- AES block would be listening to port 800000300xH – 8000003FFxH

AES Block: I/O


Select
PSELx
Strobe
PENABLE
Address and control
PADDR
PWRITE
APB slave
Reset
PRESETn
Clock
PCLK
Read data
Write data
PWDATA
PRDATA
Arbiter grant
HGRANTx
Transfer response
HREADY
HRESP[1:0]
Reset
HRESETn
Clock
HCLK
AHB master
Data
HRDATA[31:0]
HBUSREQx
HLOCKx
Arbiter
HTRANS[1:0]
Transfer type
HADDR[31:0]
HWRITE
HSIZE[2:0]
HBURST[2:0]
HPROT[3:0]
Address and control
HWDATA[31:0]
Data

AMBA bus signals (AHB Bus)

- HBUSREQx – Master requesting the bus
- HGRANTx – Indicates that bus master requesting access is the highest priority master
- HREADY – Indicates that transfer has finished
- HWRITE – When high indicates a write transfer
- HRDATA[31:0] – Read Data bus
- HWDATA[31:0] – Write data bus

AMBA bus signals (APB bus)

- PSELx – Indicates that the slave is selected
- PENABLE – Its rising edge occurs in the middle of the APB transfer
- PWRITE – When high, indicates an APB write access
- PADDR[31:0] – APB address bus
- PWDATA – APB Write data bus
- PRDATA – APB Read data bus

Steps for adding IP block(AES)

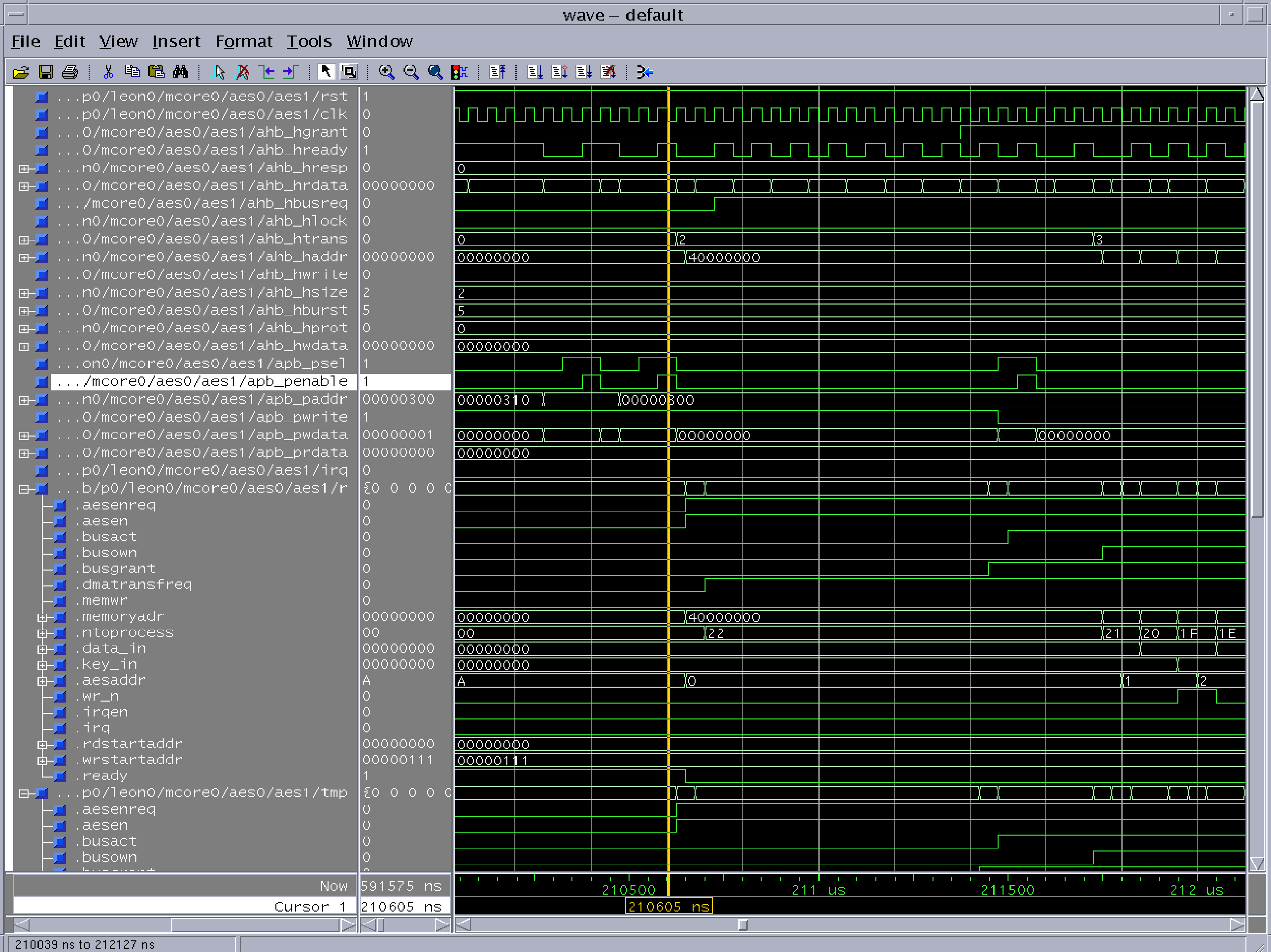
- A wrapper is required to enable communication between AES and AMBA buses (aes.vhd , aes_ctrl.vhd)
- Files to be modified: DEVICE.VHD, TARGET.VHD, MCORE.VHD,TECH_TSMC25.VHD, AMBACOMP.VHD.
- TECH_TSMC25.VHD: replace behavioral models with corresponding technology specific models.

contd….

- AMBACOMP.VHD: Add IP block as a component so that bus controller know about new AHB Bus master.
- DEVICE.VHD: specify registers which IP block would be listening to. Specify number of masters on AHB.
- MCORE.VHD: any peripheral on AHB/APB has to be added in MCORE.VHD.

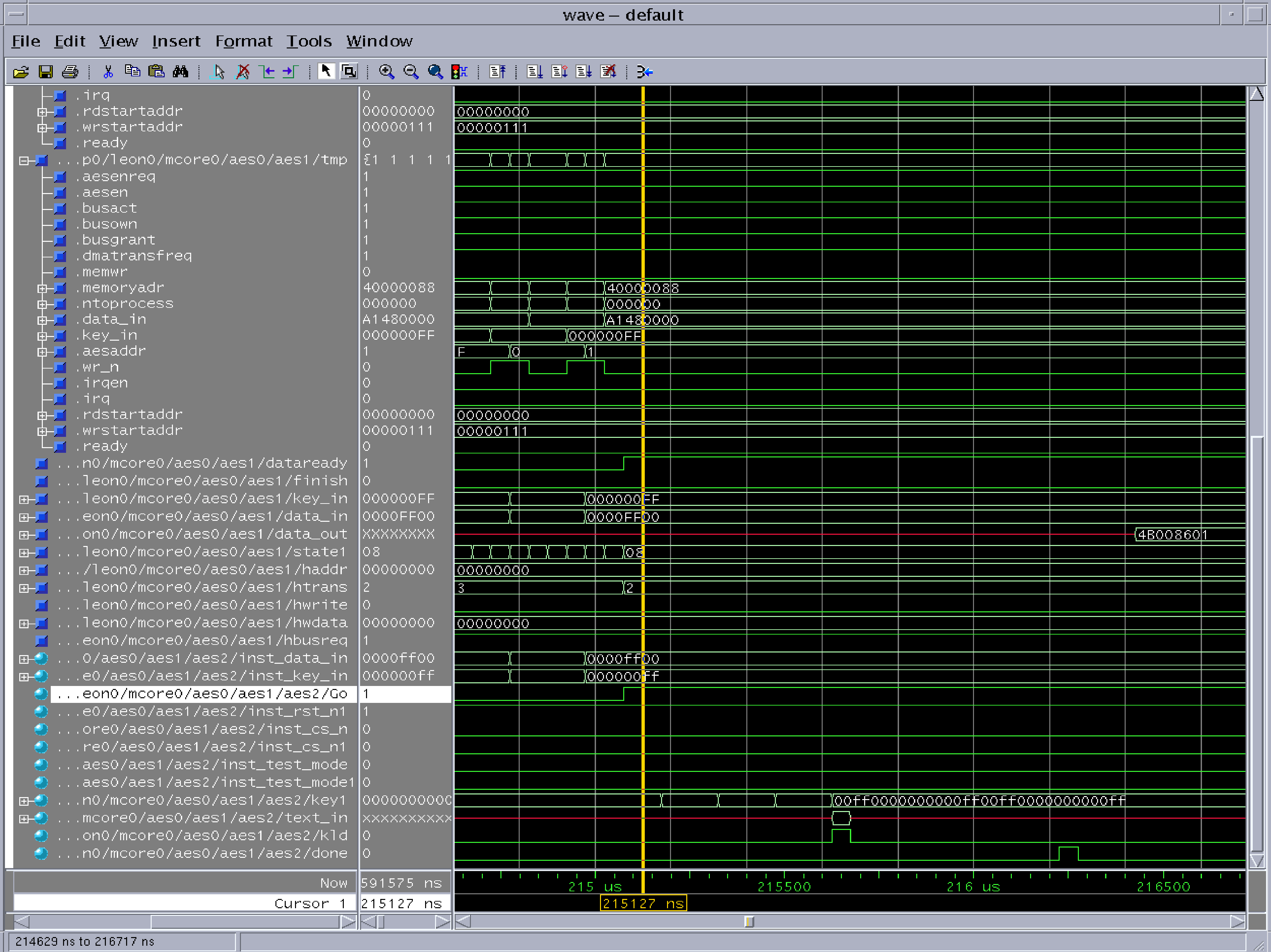
Synthesis:

- Before synthesizing Integrated SoC it is a good idea to synthesize the LEON processor – standalone.
- Generate black-box for RAM modules.
- Perform Post-synthesis simulation on the netlist using the test bench provided with LEON model.
- Synthesized netlist operated at max speed of 20Mhz.
- Finally Simulate the LEON processor with AES block and observe the waveform.



...p0/leon0/mcore0/aes0/aes1/rst	1
...p0/leon0/mcore0/aes0/aes1/clk	0
...0/mcore0/aes0/aes1/ahb_hgrant	0
...0/mcore0/aes0/aes1/ahb_hready	1
...n0/mcore0/aes0/aes1/ahb_hresp	0
...0/mcore0/aes0/aes1/ahb_hrdata	00000000
.../mcore0/aes0/aes1/ahb_hbusreq	0
...n0/mcore0/aes0/aes1/ahb_hlock	0
...0/mcore0/aes0/aes1/ahb_htrans	0
...n0/mcore0/aes0/aes1/ahb_haddr	00000000
...0/mcore0/aes0/aes1/ahb_hwrite	0
...n0/mcore0/aes0/aes1/ahb_hsize	2
...0/mcore0/aes0/aes1/ahb_hburst	5
...n0/mcore0/aes0/aes1/ahb_hprot	0
...0/mcore0/aes0/aes1/ahb_hwdata	00000000
...on0/mcore0/aes0/aes1/apb_psel	1
.../mcore0/aes0/aes1/apb_penable	1
...n0/mcore0/aes0/aes1/apb_paddr	00000300
...0/mcore0/aes0/aes1/apb_pwrite	1
...0/mcore0/aes0/aes1/apb_pwdata	00000001
...0/mcore0/aes0/aes1/apb_prdata	00000000
...p0/leon0/mcore0/aes0/aes1/irq	0
...b/p0/leon0/mcore0/aes0/aes1/r	{0 0 0 0 0
.aesenreq	0
.aesen	0
.busact	0
.busown	0
.busgrant	0
.dmatransfreq	0
.memwr	0
.memoryadr	00000000
.ntoprocess	00
.data_in	00000000
.key_in	00000000
.aesaddr	A
.wr_n	0
.irqen	0
.irq	0
.rdstartaddr	00000000
.wrstartaddr	00000111
.ready	1
...p0/leon0/mcore0/aes0/aes1/tmp	{0 0 0 0 0
.aesenreq	0
.aesen	0
.busact	0
.busown	0
Now	591575 ns
Cursor 1	210605 ns

210039 ns to 212127 ns



```
  .irq                                   0
  .rdstartaddr                           00000000
  .wrstartaddr                           00000111
  .ready                                 0
...p0/leon0/mcore0/aes0/aes1/tmp         {1 1 1 1 1
  .aesenreq                              1
  .aesen                                 1
  .busact                                1
  .busown                                1
  .busgrant                              1
  .dmatransfreq                          1
  .memwr                                 0
  .memoryadr                             40000088
  .ntoprocess                            000000
  .data_in                               A1480000
  .key_in                                000000FF
  .aesaddr                               1
  .wr_n                                  0
  .irqen                                 0
  .irq                                   0
  .rdstartaddr                           00000000
  .wrstartaddr                           00000111
  .ready                                 0
...n0/mcore0/aes0/aes1/dataready         1
...leon0/mcore0/aes0/aes1/finish         0
...leon0/mcore0/aes0/aes1/key_in         000000FF
...eon0/mcore0/aes0/aes1/data_in         0000FF00
...on0/mcore0/aes0/aes1/data_out         XXXXXXXX
...leon0/mcore0/aes0/aes1/state1         08
.../leon0/mcore0/aes0/aes1/haddr         00000000
...leon0/mcore0/aes0/aes1/htrans         2
...leon0/mcore0/aes0/aes1/hwrite         0
...leon0/mcore0/aes0/aes1/hwdata         00000000
...eon0/mcore0/aes0/aes1/hbusreq         1
...0/aes0/aes1/aes2/inst_data_in         0000ff00
...e0/aes0/aes1/aes2/inst_key_in         000000ff
...eon0/mcore0/aes0/aes1/aes2/Go         1
...e0/aes0/aes1/aes2/inst_rst_n1         1
...ore0/aes0/aes1/aes2/inst_cs_n         0
...re0/aes0/aes1/aes2/inst_cs_n1         0
...aes0/aes1/aes2/inst_test_mode         0
...aes0/aes1/aes2/inst_test_mode1        0
...n0/mcore0/aes0/aes1/aes2/key1         000000000C
...mcore0/aes0/aes1/aes2/text_in         xxxxxxxxxx
...on0/mcore0/aes0/aes1/aes2/kld         0
...n0/mcore0/aes0/aes1/aes2/done         0

Now        591575 ns
Cursor 1   215127 ns

214629 ns to 216717 ns
```

References

- Rishi's Thesis : http://vlsi1.engr.utk.edu/ece/rishi-thesis.pdf
- AMBA documentation : http://www.gaisler.com/doc/amba.pdf
- LEON-2 processor User's Manual : http://www.gaisler.com
- Artisan Components,Inc. "Generator User Manual"

Special Thanks to Wei Jiang for his help !

Questions?

Thank You!