MIPS Integrated Architectural Memory Design Synthesis for Low Power Embedded Devices

Ravi Khatwal
Senior researcher
Department of computer science
Mohan Lal Sukhadia University
Udaipur, India

Manoj Kumar Jain
Associate professor
Department of computer science
Mohan Lal Sukhadia University
Udaipur, India

ABSTRACT

Recently high performance and low power consumption custom memory design system is the crucial innovation for wireless embedded devices. In this paper we have implemented MIPS based memory architectural design and analyze its simulation efficiency. Low power and high performance embedded devices such as mobile, wifi devices implemented with MIPS architecture design that reduces the access time and increases the system performance. Existing CAM cell design also reduces the access time as in efficient manner. We have analyzed CAM architecture design with xup-5 FPGA environments and analyze CAM cell efficiency. MIPS RF memory has implemented for various high performances ASIP design architecture and embedded devices.

Keywords

MIPS RF design, ASIP Simulation, Xup-5 FPGA device, Xilinx, Simulation design.

1. INTRODUCTION

In HDL simulation, Xilinx simulator software verifies the functionality and timing of MIPS design architecture. This simulator used to interpret the VHDL or Verilog code into circuit functionality and performs the logical results of HDL to determine the correct our circuit operation. Xilinx simulator is used to creating and verifying the complex functions in a relatively small amount of time. Jain, M.K., Balakrishnan M. and A. Kumar A. [1] designed scheduler based technique for exploring register file size; number of register window and cache configuration. Kuldar, M., Fan, K., Chu M. and Mahlke, S. [2] proposed a technique to synthesize the local memory architecture of a clustered accelerator using a phase-ordered approach. Vahid, F., Stitt, G., Guo Z. [3] proposed synthesis method for FPGAs utilize advanced memory structure, such as smart buffer, that require recovery of additional high-level information about loops and array. P. Meloni, S. Pomato, G. Tuveri, L. Raffo, M. Lindwer, [5] proposed an emulation tool chain for ASIP design. Prikryl, Z., Kroustek, I., Hruska, T., Kolar, D. [6] proposed fast and accurate processor simulator is used for effective design of modern high performance application specific instruction set processor. Meloni, P., Pomato, S., Raffo, L., Lindwer, M. [7] proposed integrated tool chain design & evaluated the ASIP systems. Jordans, R., Diken, E., Jozwiak, L., Corporaul, H. [8] proposed build master framework. This framework used efficient automated cache compilation and simulation for ASIP. C.K. Lai, Y.J. Huang and J.F. Lia [9] proposed a simple and effective built-in self-repair (BISR) scheme for content addressable memories (CAMs) with address-input-free writing function. Three novel cache models [10] using Multiple-Valued Logic (MVL) to reduce the cache data storage area and cache energy consumption for embedded systems. Spin-transfer torque RAM (STT-RAM) [11] is an emerging nonvolatile memory technology that has low-power and high-density advantages over SRAM.

2. MIPS REGISTER FILE DESIGN

MIPS 64-bit memory architecture (see figure 1) design implemented when clock enabled condition input and IRWrite High, and the synchronous reset (Rst_n) input is Low, the data inputs (memory Data) is transferred to the corresponding data outputs (instruction) during the Low-to-High clock (clk) transition. When Rst_n is high, resets the data output Low on the Low-to-High clock transition states. When IRWrite is Low, there is no transitions occurred. MIPS memory architecture performs efficient memory operations.

2.1 Instruction Implemented 64-bit MIPS RF

MIPS perform instruction set designs have implemented the output form data in long and short instruction format. Synthesize design have analyzed the MIPS RF and viewing its internal structure (see figure 2&3) with XUP-5 FPGA environment. Input design section has implemented in the form of memory data When IRWR activated memory write operation is performed and memory read operation performed data output state. 16-bit output instruction design performs the internal operations of ISA level. After doing these simulations process access time and delay time also reduce.

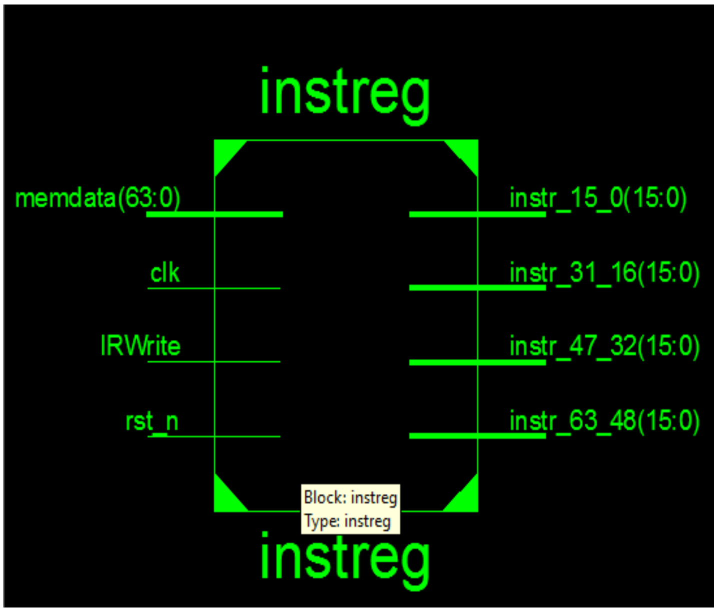


Fig 1: Instruction implemented MIPS RF design

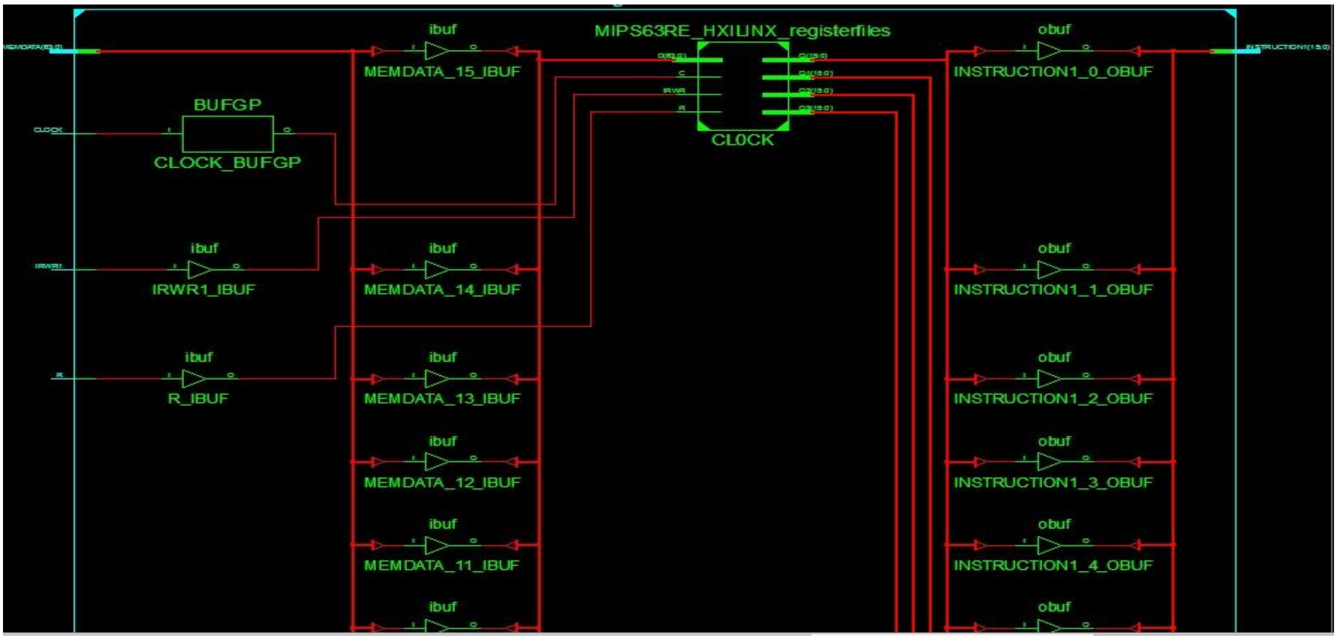
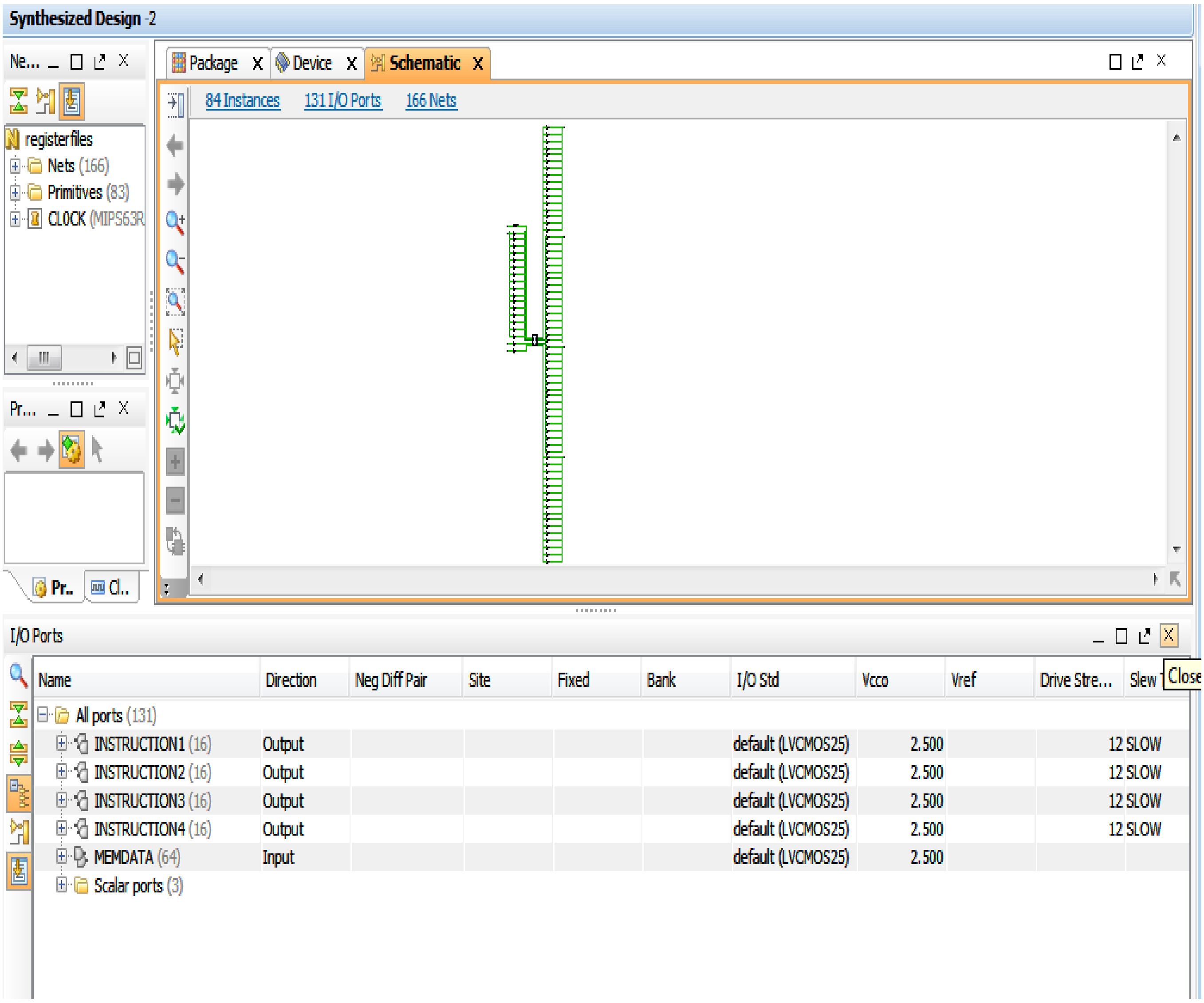


Fig 2: RTL design of MIPS RF design



Name	Direction	Neg Diff Pair	Site	Fixed	Bank	I/O Std	Vcco	Vref	Drive Stre...	Slew
All ports (131)										
INSTRUCTION1 (16)	Output					default (LVCMOS25)	2.500		12	SLOW
INSTRUCTION2 (16)	Output					default (LVCMOS25)	2.500		12	SLOW
INSTRUCTION3 (16)	Output					default (LVCMOS25)	2.500		12	SLOW
INSTRUCTION4 (16)	Output					default (LVCMOS25)	2.500		12	SLOW
MEMDATA (64)	Input					default (LVCMOS25)	2.500			
Scalar ports (3)										

Fig 3: Synthesize 64-bit MIPS register file design

2.2 FPGA Implementation of MIPS RF Design

FPGA boards allow as viewing and verifying the post-Place, Route implementation of our design. During these simulation FPGA board allow as to directly modifying the physical design architecture and implement the UCF files to verify our design. Memory data, memory hierarchy, clocks and instruction design analyzed in XUP-5 FPGA environments (see figure 4, 5, 6& 7). MIPS RF design have implemented on xup-5 FPGA device (see figure 8) and simulation result implemented for low power embedded devices. MIPS RF VHDL OR VERILOG code analyzes the logical results of the HDL to determine internal circuit operations. During HDL analysis, XST used to check the circuits HDL code is correct or not. Input memory data produces the output instruction formats and these designs perform less access time and propagation delay time. MIPS RF design code performs better simulation results (see figure 9) and these design consume less access time so it can be used for high performances devices.

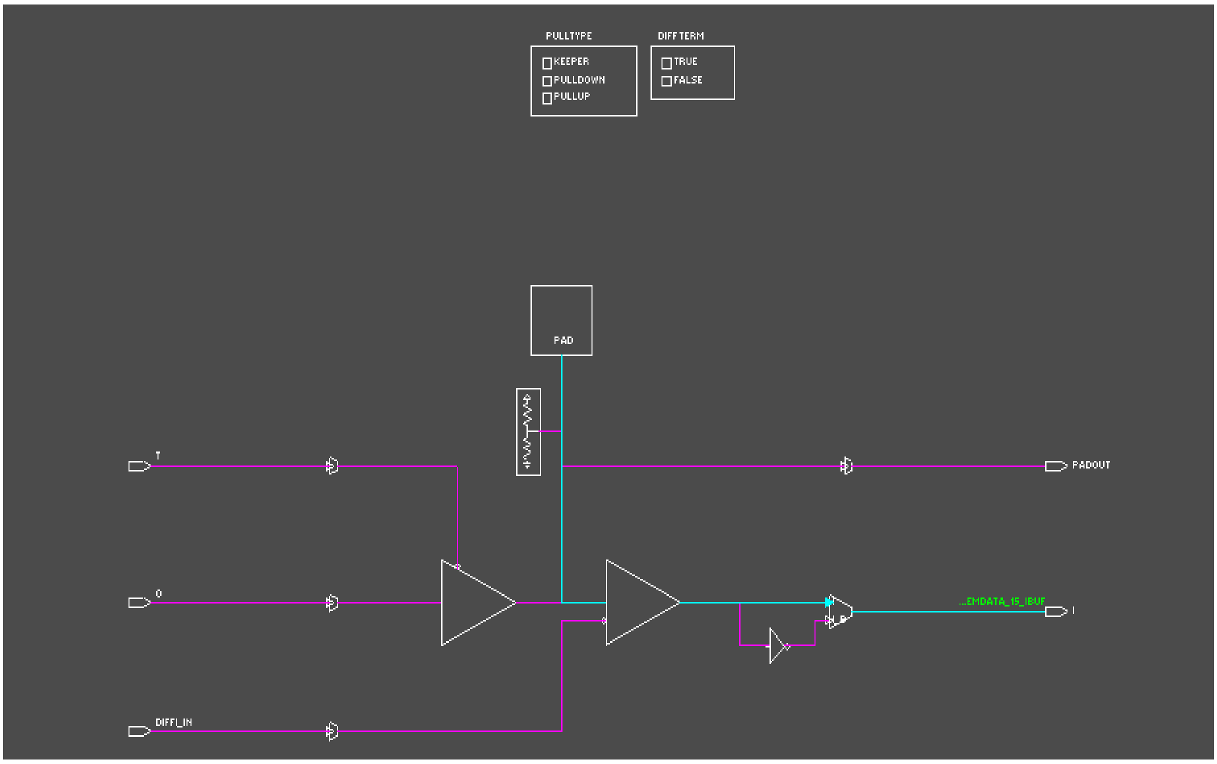


Fig 4: Memory Data analysis in FPGA

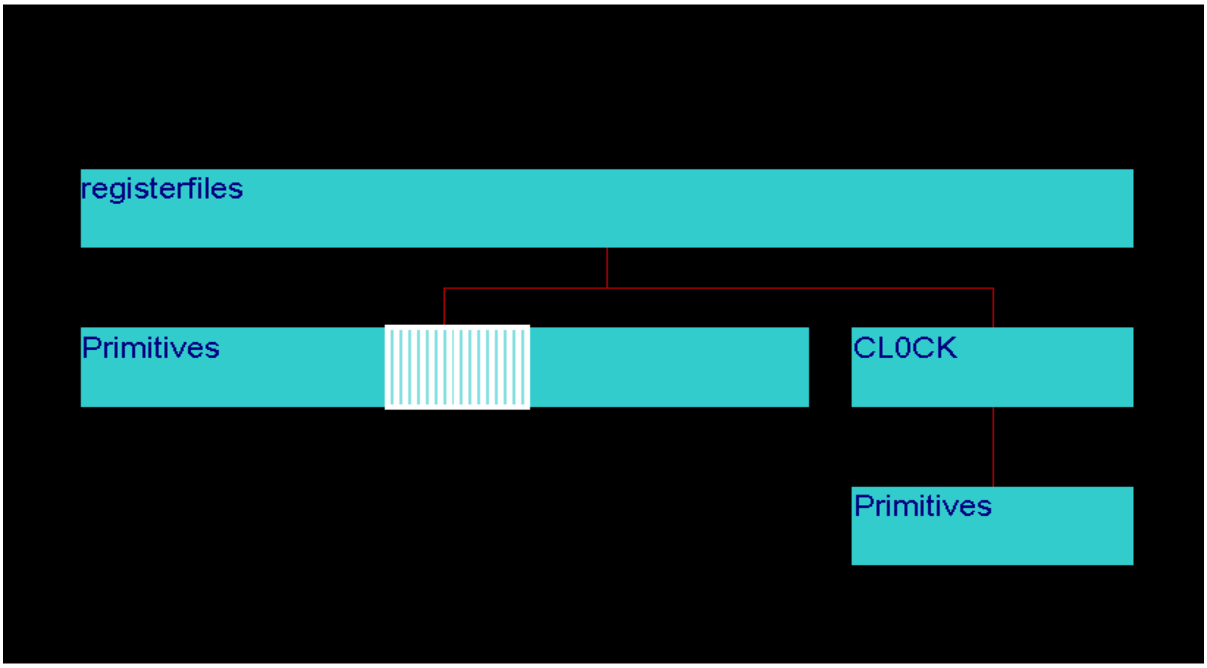


Fig 5: Hierarchy design of MIPS

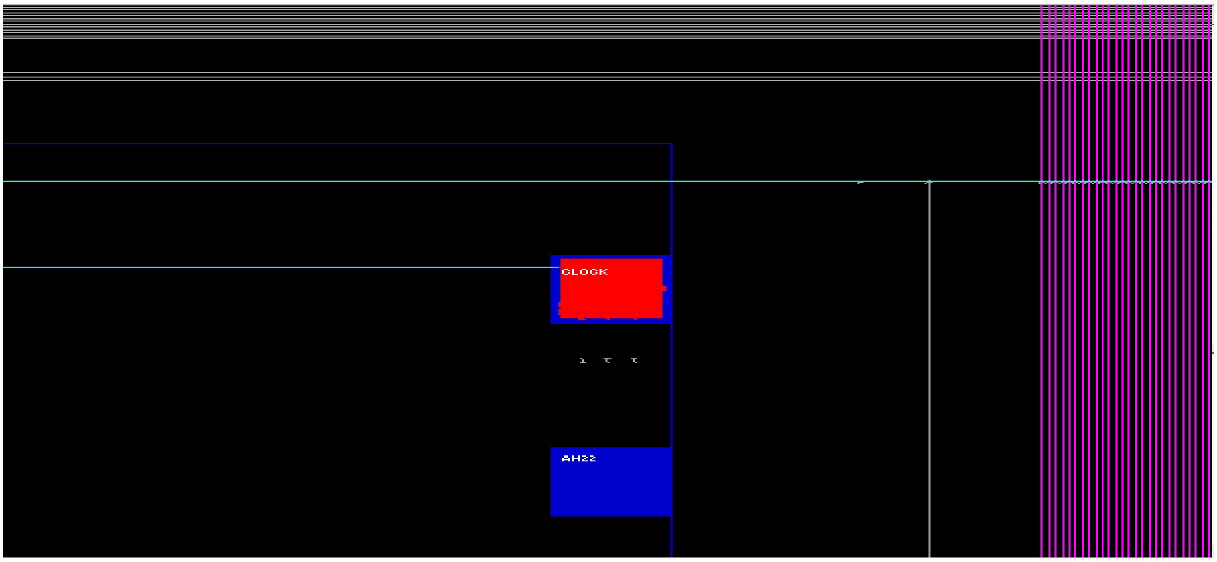


Fig 6: Clock analysis in XUP-5 FPGA

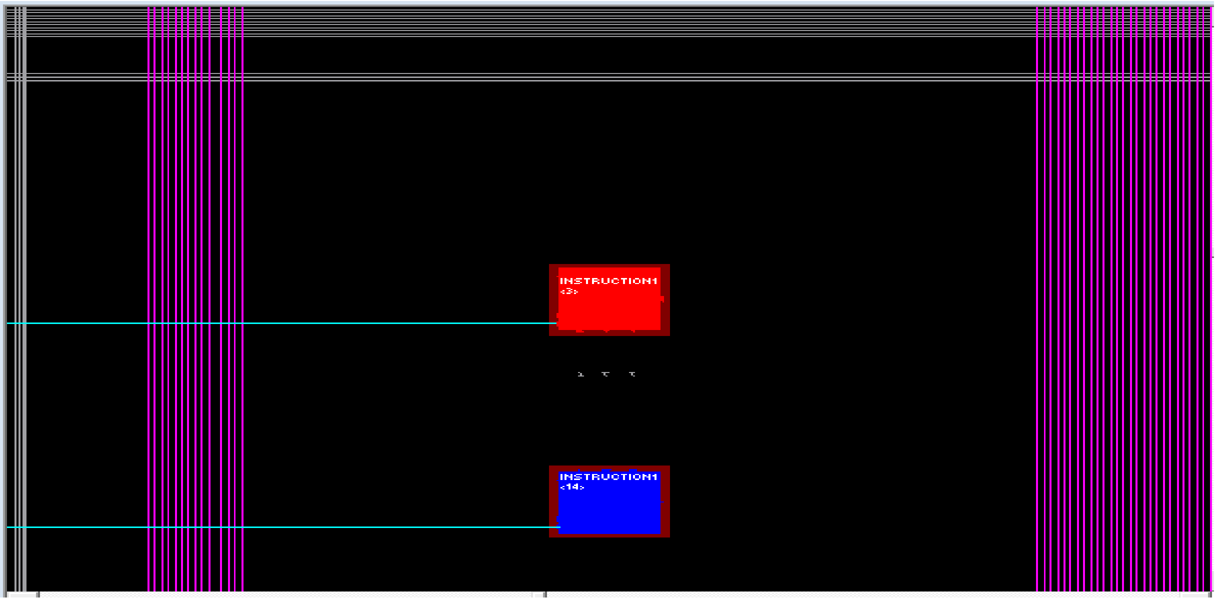
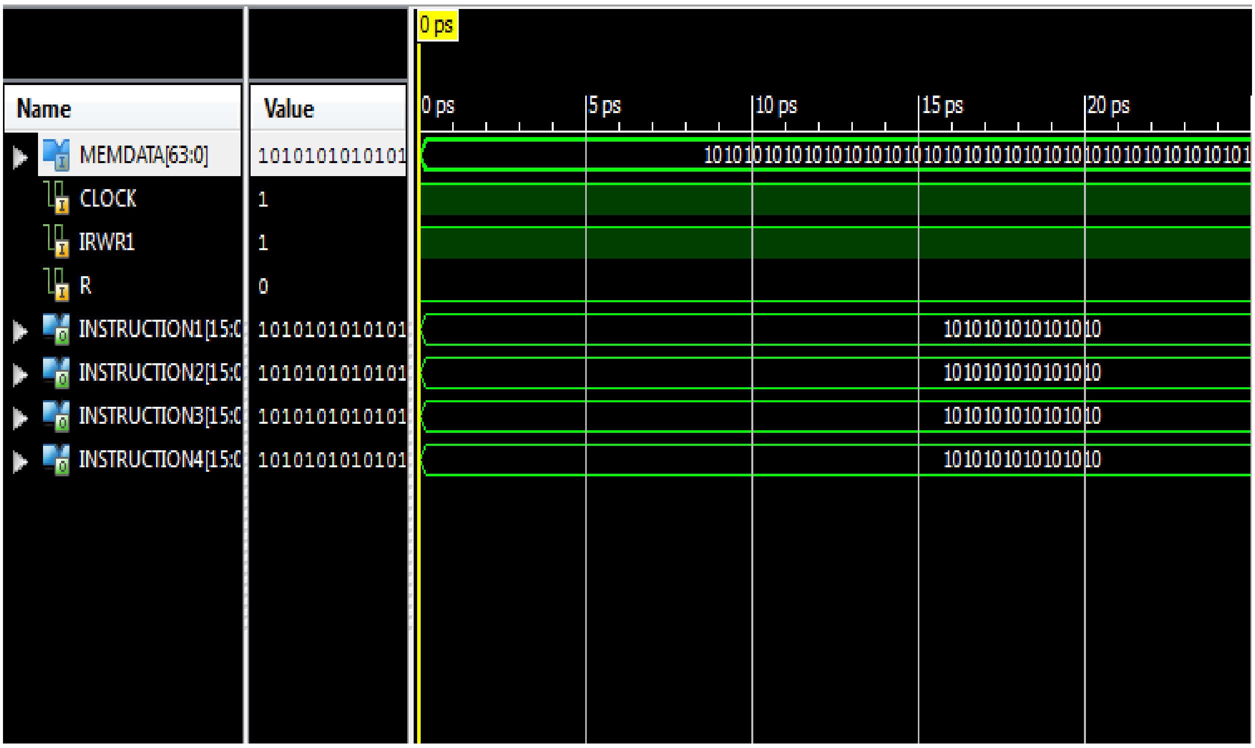


Fig 7: Instruction analysis in xup-5 FPGA Board

Fig 8: Xup-5 board



Name	Value
MEMDATA[63:0]	1010101010101
CLOCK	1
IRWR1	1
R	0
INSTRUCTION1[15:0	1010101010101
INSTRUCTION2[15:0	1010101010101
INSTRUCTION3[15:0	1010101010101
INSTRUCTION4[15:0	1010101010101

Fig 9: Simulation results

3. CAM CELL DESIGN

CAM (see figure 10) cell perform its operation when match circuits is in active state. Each CAM cell has match logic as comparator to check the match contents. When match activated, indicating a match occur, while the other match lines indicating a mismatch condition. During CAM simulation process we have analyzed the synthesis design architecture with XUP-5 FPGA environment. We implements low levels CAM design with 1024 data input state and output addresses in 8-bit forms respectively. CAM design implemented with match logic as comparator, data input circuits, address circuits, output circuits etc (see figure 11). Synthesize design architecture perform the matching operation and have simulation access time as 0.1ns due its design complexity (see figure 12& 13). When CAM hit occur then CAM enable & CLK unit activated and performs cam output results. After this simulation we get lower propagation delay time.

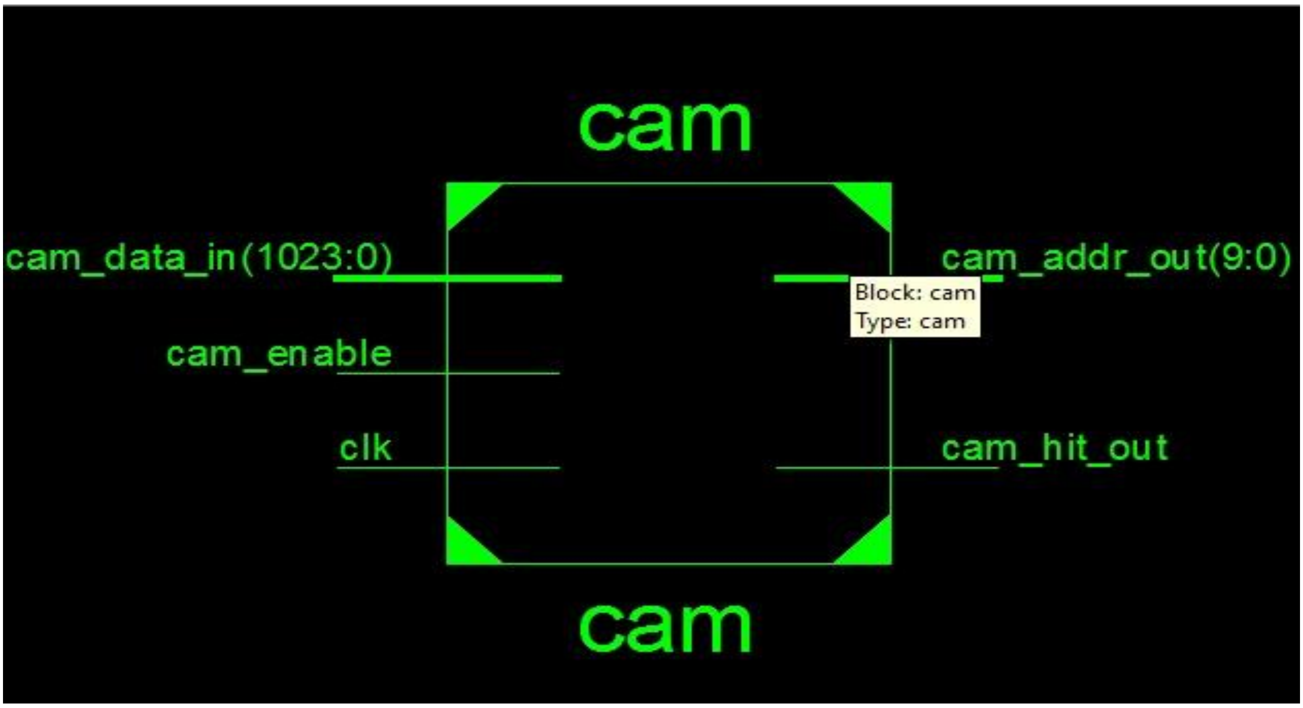
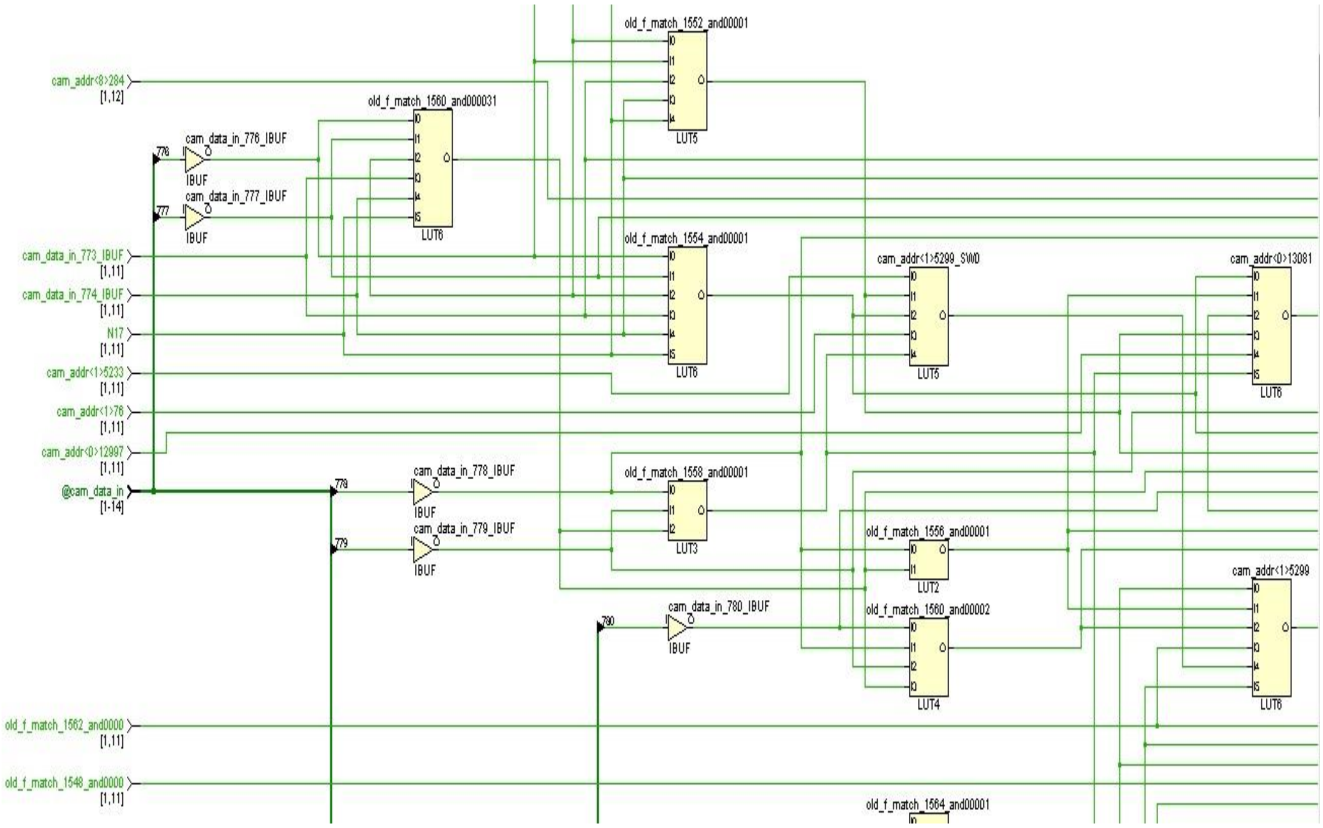
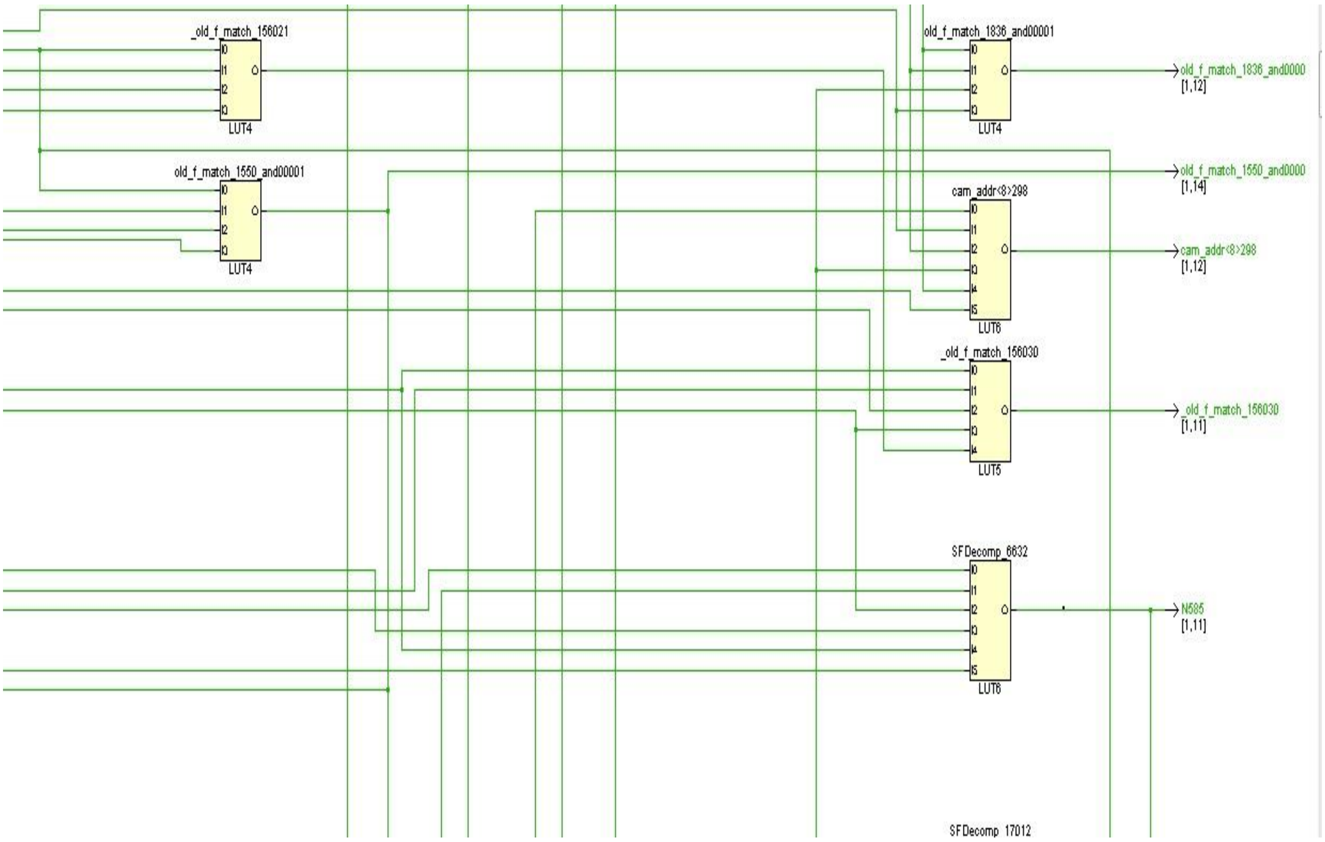


Fig 10: Cam design analysis



(a) CAM input section design



(a) CAM output section design

Fig 11: Cam design architecture

(a) CAM input section design & (b) CAM output section design

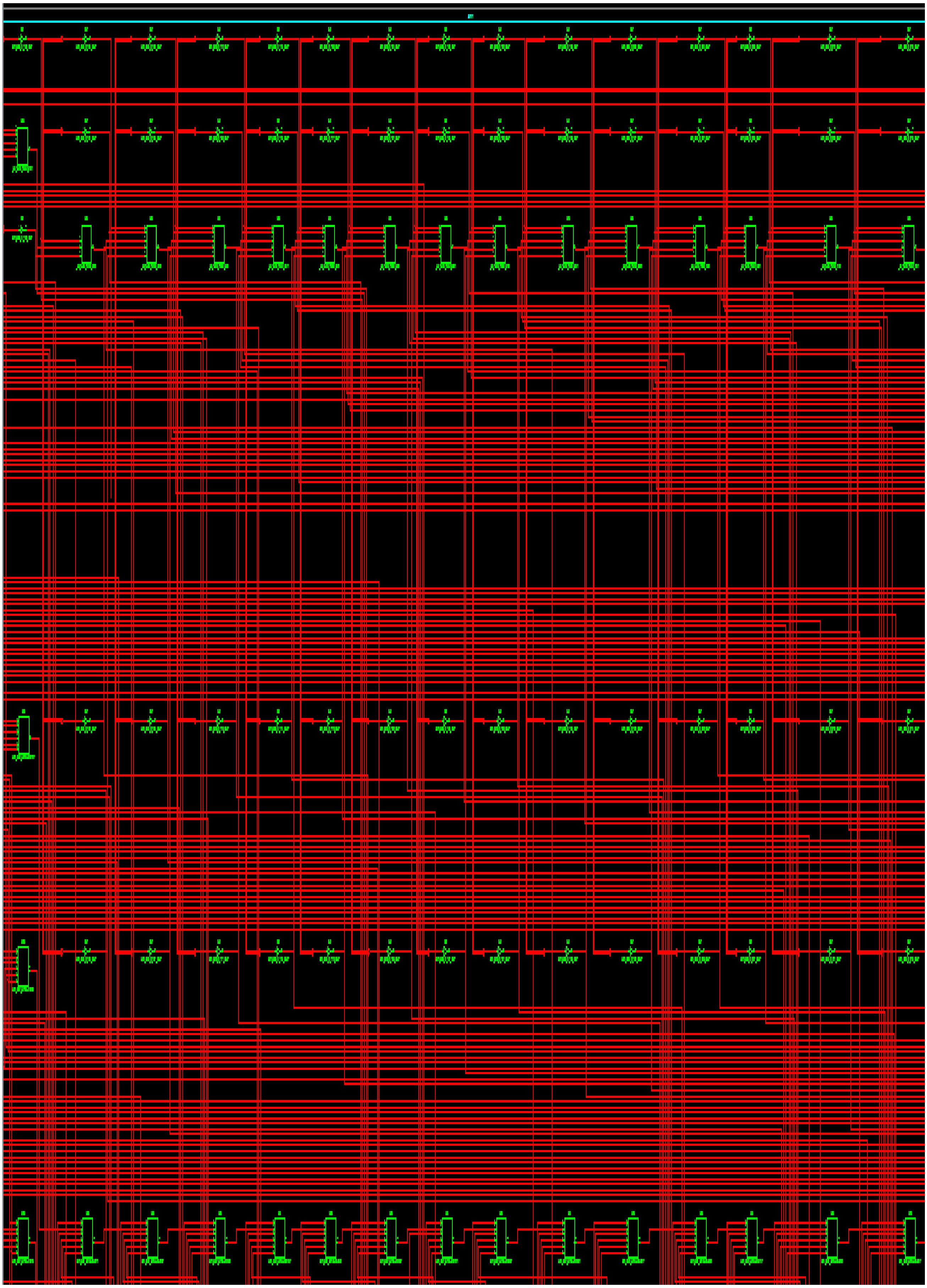


Fig 12: CAM RTL Design

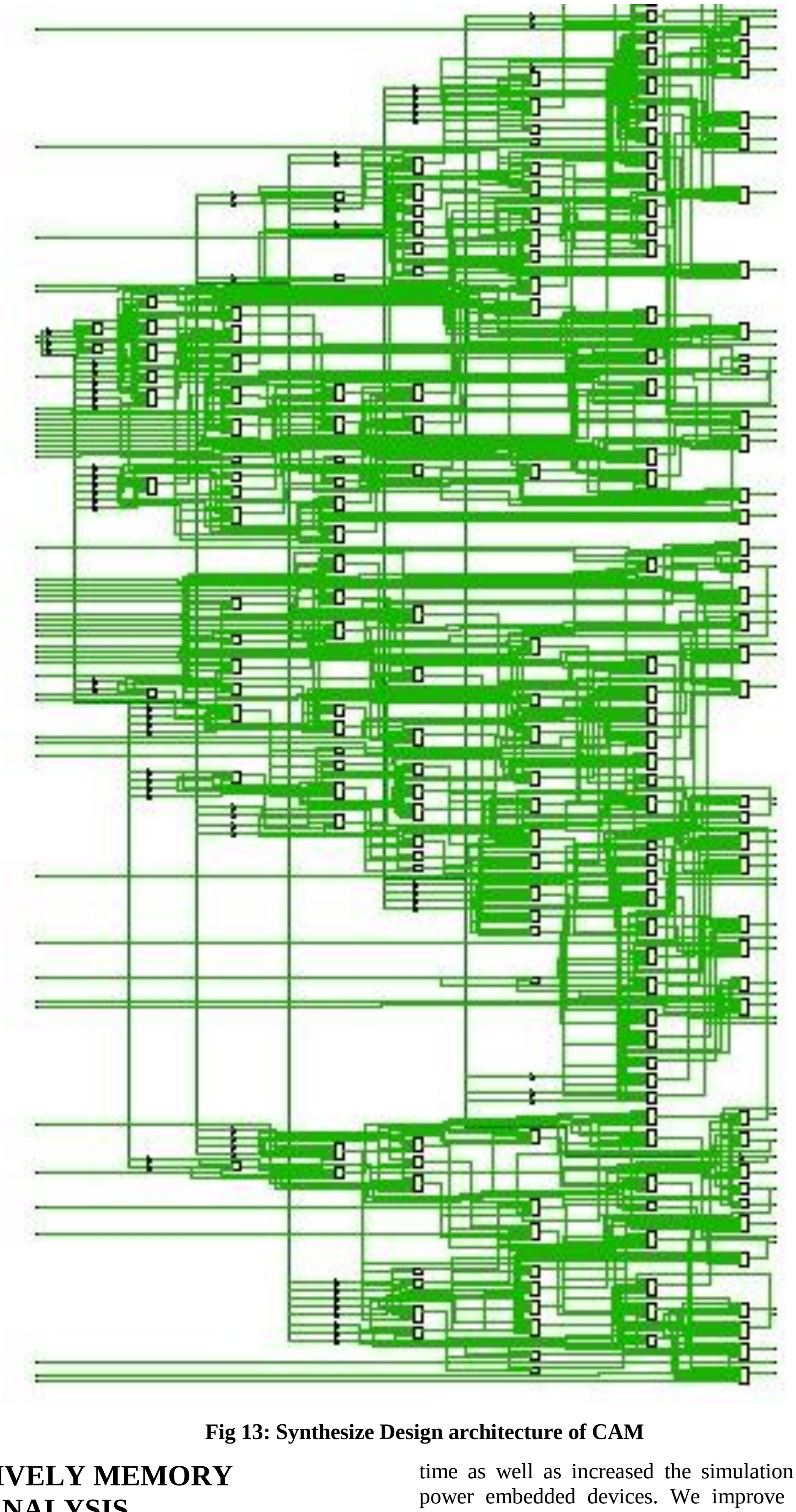
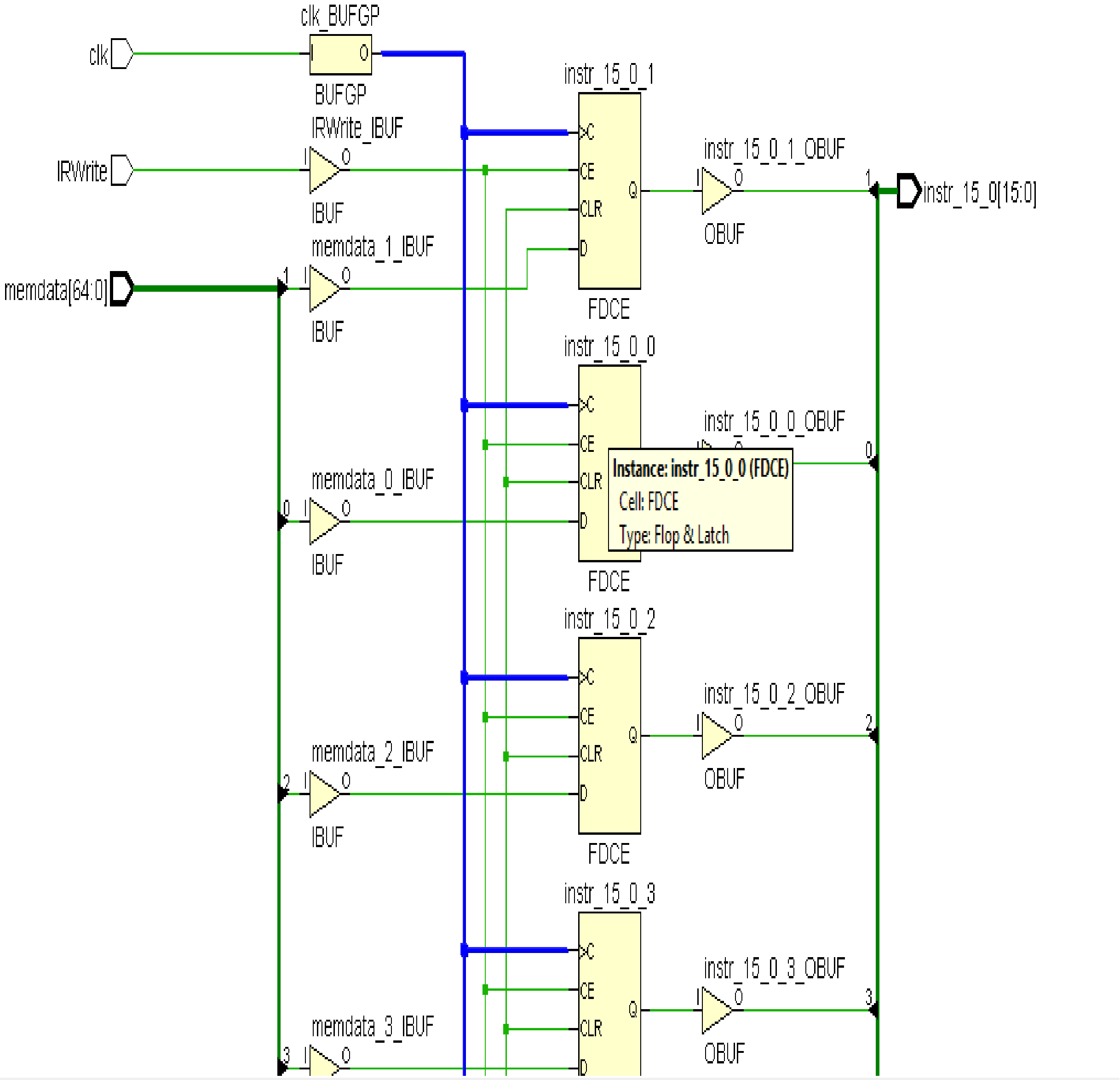


Fig 13: Synthesize Design architecture of CAM

4. COMPARATIVELY MEMORY SIMULATION ANALYSIS

In comparative memory simulation procedure recently cache memory performs accurate, low power consuming & have lower access time it became cheaper than cam and RF. MIPS based memory architecture memory design implemented that performs efficient simulation results and reduces the access time as well as increased the simulation efficiency for low power embedded devices. We improve memory efficiency with internal clock mechanism (see figure 14). Comparatively analyze that SRAM have access time as 1ns and CAM cell contains memory access time as 0.1ns due to its design complexity but MIPS implemented architecture memory deign reduces the propagation delay time and have access time as 0.01ns (see figure 15).



Instance: instr_15_0_0 (FDCE)
Cell: FDCE
Type: Flop & Latch

Fig 14: Internal clock mechanism of MIPS RF design

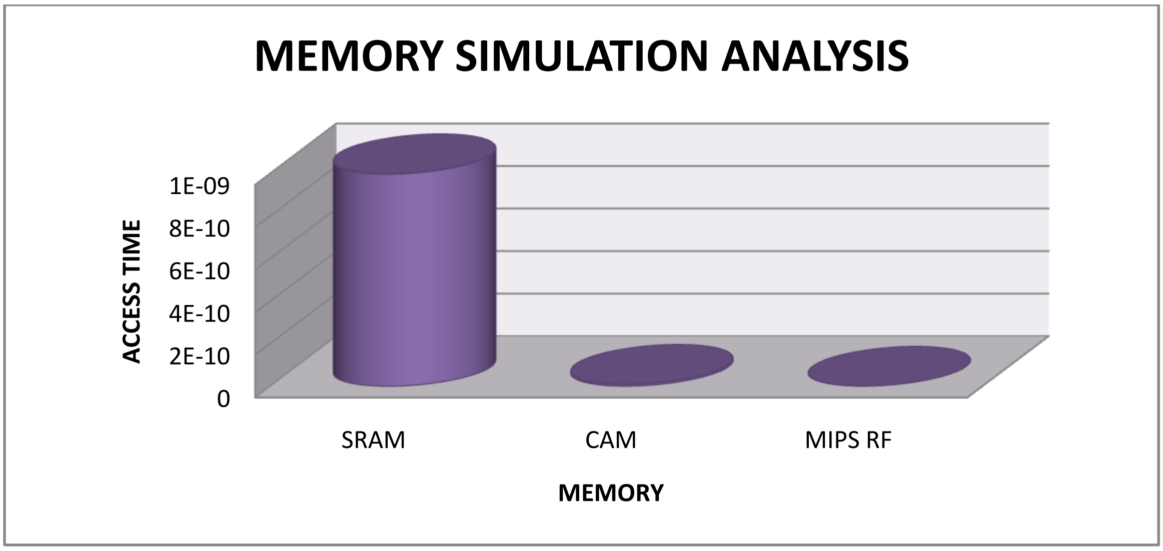
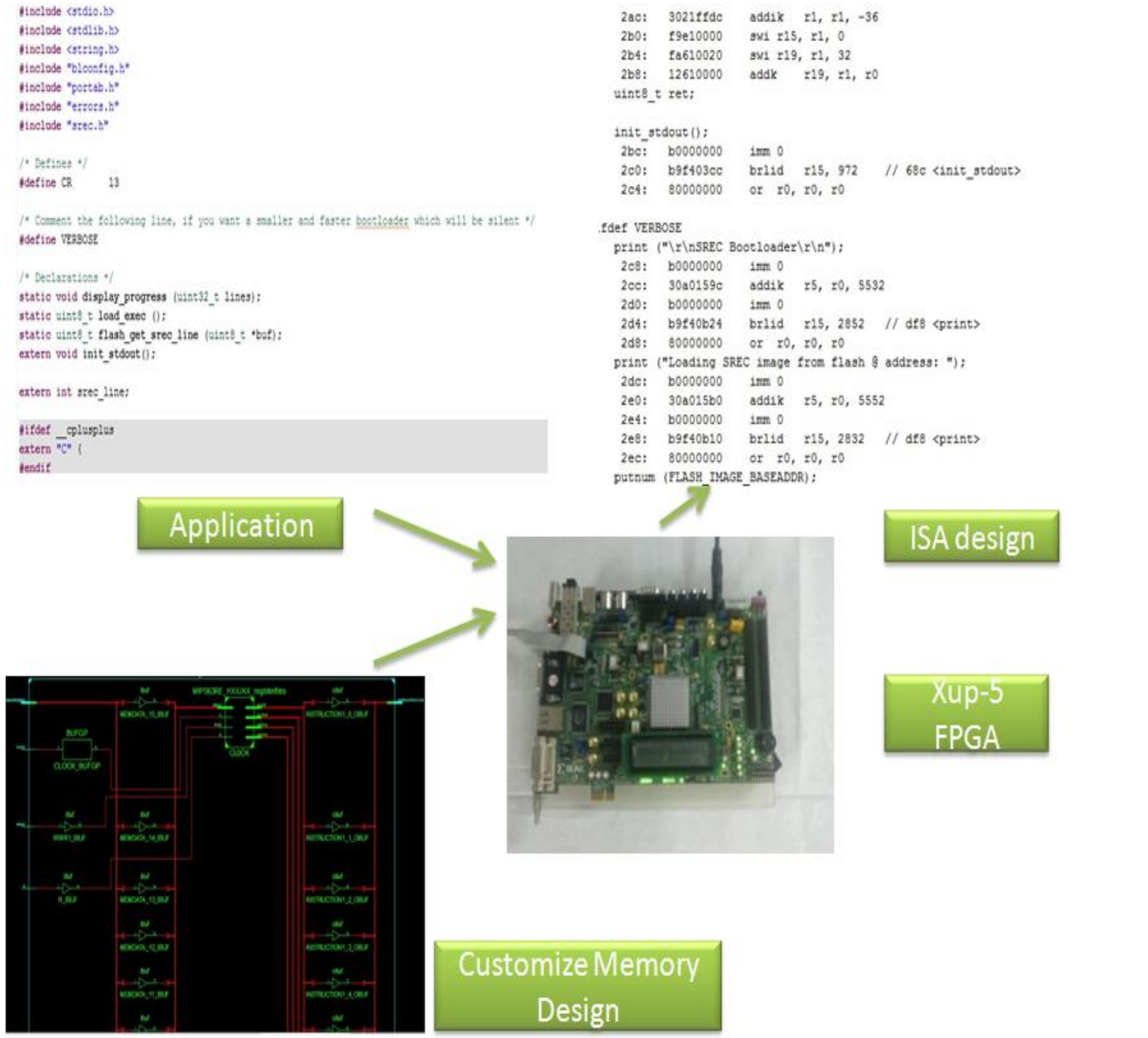


Fig 15: Comparitive analysis of MIPS register file

5. APPLICATION SPECIFIC ARCHITECTURAL MEMORY DESIGN SIMULATION

Memory architectural design have implemented according to specific application. We have used standard boot loader application and integrate the MIPS RF memory architecture on targeted hardware environment. After targeting we have analyzed the ISA behaviour for specific application. These simulations processes ISA architecture behaviour analyzed for specific application with XUP-5 FPGA [4] hardware platform (see figure 16). Recently MIPS RF memory design implemented for ASIP design architecture and low power Embedded designing.



```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "blconfig.h"
#include "portab.h"
#include "errors.h"
#include "srec.h"

/* Defines */
#define CR       13

/* Comment the following line, if you want a smaller and faster bootloader which will be silent */
#define VERBOSE

/* Declarations */
static void display_progress (uint32_t lines);
static uint8_t load_exec ();
static uint8_t flash_get_srec_line (uint8_t *buf);
extern void init_stdout();

extern int srec_line;

#ifdef __cplusplus
extern "C" {
#endif
```

```
   2ac:   3021ffdc    addik   r1, r1, -36
   2b0:   f9e10000    swi r15, r1, 0
   2b4:   fa610020    swi r19, r1, 32
   2b8:   12610000    addk    r19, r1, r0
  uint8_t ret;

  init_stdout();
   2bc:   b0000000    imm 0
   2c0:   b9f403cc    brlid   r15, 972    // 68c <init_stdout>
   2c4:   80000000    or  r0, r0, r0

.fdef VERBOSE
  print ("\r\nSREC Bootloader\r\n");
   2c8:   b0000000    imm 0
   2cc:   30a0159c    addik   r5, r0, 5532
   2d0:   b0000000    imm 0
   2d4:   b9f40b24    brlid   r15, 2852   // df8 <print>
   2d8:   80000000    or  r0, r0, r0
  print ("Loading SREC image from flash @ address: ");
   2dc:   b0000000    imm 0
   2e0:   30a015b0    addik   r5, r0, 5552
   2e4:   b0000000    imm 0
   2e8:   b9f40b10    brlid   r15, 2832   // df8 <print>
   2ec:   80000000    or  r0, r0, r0
  putnum (FLASH_IMAGE_BASEADDR);
```

Fig 16: ASIP simulation analysis with MIPS RF

6. CONCLUSIONS

MIPS memory architecture design analyzed with the help of Xilinx simulator. MIPS RF design reduces the access time with high speed microprocessor clock rate. We have analyzed the MIPS memory efficiency and verified our design with xup-5 FPGA environments. After doing this simulation process we can get efficient result regarding highly reduction of simulation access time used for recently artificial intelligence, mobile, wireless and medical devices. These MIPS implemented memory design performs efficient ISA design architecture and performs lower access time. Comparatively analysis of MIPS based memory architecture performs less propagation delay time so it can be used for ASIP design architecture and future high performance design implementation.

7. REFERENCES

[1] Jain, M.K., Balakrishnan, M. and Kumar A., 2005.Integrated on-chip storage evaluation in ASIP synthesis, In Proceeding of 18th International Conference on VLSI design, 274 – 279.

[2] Kuldar, M., Fan, K., Chu, M. and Mahlke, S. 2004.Automatic Synthesis of Customized Local Memories for Multiclustered Application Accelerators, In Proceeding of 15th IEEE International Conference on Application –Specific Systems, Architectures and Processors.304-314.

[3] Vahid, F., Stitt, G., Guo, Z., Najjar, W.2005. Technique for Synthesizing Binaries to an advanced

Register/Memory Structure, SIGA 13th International symposium on FPGA (2005).118-124.

[4] Xilinx tool [online] Available: www.xilinx.com/homepage/.

[5] P. Meloni, S. Pomato, G. Tuveri, L. Raffo, M. Lindwer, "Enabling fast ASIP design space exploration: An FPGA based runtime reconfigurable prototype". Hindawi Publication Cooperation Journal of VLSI design (2012).

[6] Prikryl, Z., Kroustek, I., Hruska, T., Kolar, D. Fast just in time translated simulator for ASIP. Design and diagnostics of electronic circuits and system (DDECS), 2011, IEEE 14TH international symposium. 279-282.

[7] P.Meloni, S.Pomato, R.Piscitelli, L.Raffo, M.Lindwer. Combining on-hardware prototyping and high level simulation for DSE of multi-ASIP system. IEEE Embedded Computer Systems (SAMOS), (2012).310-317.

[8] Jordans, R., Diken, E., Jozwiak, L., Corporaul, H. (2014).Build master: efficient ASIP architecture exploration through compilation and simulation result caching. IEEE 17th International Symposium on Design and Diagnostics of Electronic Circuits & Systems,. (2014).83-88.

[9] C. K. Lai, Y.J. Huang and J.F. Lia. "Self-Repair Technique for Content Addressable Memories with Address-Input-Free Writing Function", Journal of information science and engineering. (2013).493-507.

[10] Ozer, E., Sendag, R. and Gregg, D. 2005.Multiple-Valued Caches for Power-Efficient Embedded Systems. In Proc of the 35th International Symposium on Multiple-Valued Logic (ISMVL'05) (2005).

[11] Wang, P., Sung, G., Wang, T., Xie, Y., Cong, J. (2013).Designing Scratchpad Memory Architecture with Emerging STT-RAM Memory Technologies.Circuits and system (ICACS) (2013) 1244-1247.