## Benha University

**Benha Faculty of Engineering Electrical Department**

Pipelined mips processor

By

**karem mohamed abd elmenam**

**Karem gamal eissa**

**Sherouk ahmed**

**Azza afifi**

**Farida ahmed**

**Fatma mohamed**

**Aya mohamed**

Under Supervision

Dr. Hatem zakaria

1. abstract

paper about 24-bit processors. Here is a possible introduction:

A MIPS 24-bit pipelined processor is a processor that uses the MIPS instruction set architecture (ISA) and can operate on 24-bit data units. A pipelined processor divides the execution of an instruction into several stages, such as instruction fetch, instruction decode, execute, memory, and writeback. Each stage is performed by a separate hardware unit, and multiple instructions can be processed simultaneously in different stages. This improves the throughput and performance of the processor, as long as there are no pipeline hazards, such as data dependencies, control hazards, or structural hazards.

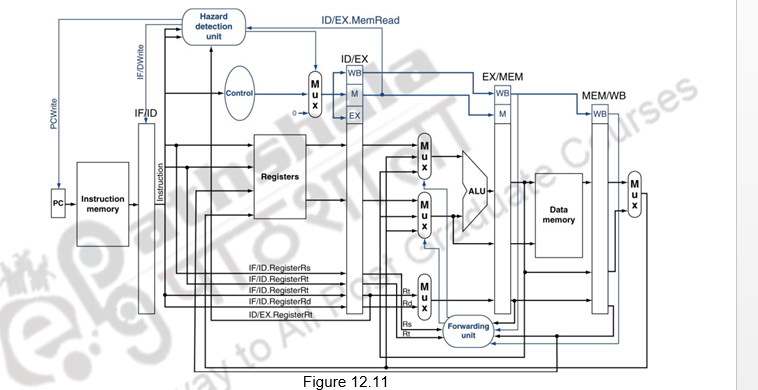
One example of a MIPS 24-bit pipelined processor is the 65816, which is a microprocessor and microcontroller family with 16-bit registers and 24-bit bank switched addressing.

1. Introduction

Computer architecture is the design and organization of the hardware and software components of a computer system. One of the key aspects of computer architecture is the word size, which is the natural unit of data used by a particular processor design. The word size determines how many bits or digits can be processed, stored, and transferred in a single operation. The word size also affects the memory size, address size, data types, performance, and compatibility of a computer system. Different word sizes have been used throughout the history of computing, ranging from 4 bits to 80 bits, depending on the needs and capabilities of the applications and devices. In this paper, we will focus on one specific word size: 24 bits. We will explore the concept and usage of 24-bit processors.

1. Pipeline concept.

MIPS ARCHITECTURE The accompanying outline demonstrates the fundamental architecture of a MIPS-based framework. Basic MIPS Architecture Microprocessor without Interlocked Pipeline Stages (MIPS) is a RISC (Reduced Instruction Set Computing) architecture. Pipelined MIPS has five stages which are IF, ID, EX, MEM and WB. Pipelining means several operations in single data path at the same instant. Pipelining is used to enhance the capabilities of the RISC processor which is the reason for its utilization in this type of computer architecture. A multicycle CPU comprises of countless tasks. So if one task occurs, rather than waiting for the process to finish, at the same time another task is initiated in the same data path simultaneously without interfering with the previous task. The processes is thus divided into different pipelined stages. Following every clock a new operation is instigated in the pipeline stage to which the process is being fed to. The triggering is done without causing any interruptions to the past process. This makes simultaneous utilization of all stages in the data path possible. This thusly can increment the throughput of MIPS. 5-Stage Pipelined MIPS



MIPS processor has been executed utilizing five pipeline stages, which are Instruction Fetch (IF), Instruction Decode (ID), Execution (EX), Memory access (MEM) and Write Back (WB).The isolation of these stages is achieved by special registers known as pipeline registers. The aim of these registers is to isolate the stages of the instructions so that there is no inadmissible information because of various directions being executed all the while. They are named in the middle of each of these: IF/ID Register, EX/MEM Register and MEM/WB Register. The data path demonstrated in Fig. 0. is that of the MIPS pipelined processor.

* 1. Instruction Fetch (IF) .

The command relayed to the Program Counter (PC) to fetch the instruction from the cache memory is what instigates the primary pipelining operation of the IF stage. The storage of PC and Instruction for the successive clock cycle is done in the IF/ID pipelined register as RAM (Random Access Memory) Fig. 1. IF Stage representation IF stage for the most part relies on upon PC’s represent value. On the basis of the PC value the processor gets the instructions from the cache and followed by which the Program Counter value is incremented by 1. Thus, the IF/ID register receives this information followed by which the information is relayed to the decoder unit. The Instruction Fetch (IF) stage operation has been represented in Fig. 1 .

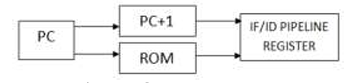


Fig1 .Instruction Fetch.

* 1. Instruction Decoder (ID)

ID Stage representation The Opcode is relayed to the decoder unit at the instant when the instruction is obtained from the IF stage. Instruction Decoder ID stage directs the controlling command to the various units of the MIPS processor examining the Opcode of the instructions. Thus the procurement of data from the MIPS registers is carried out by the Read register. The Branch unit is likewise incorporated into Instruction Decoder (ID) stage. The Input data of ID stage is received from IF stage as shown in Fig. 1.This decoding stage includes four different instructions: Register (R) type, Immediate (I)type, Jump (J) type and Input/Output (I/O) type instructions. Depending upon these instructions the function will be performed utilizing above mentioned formats. Fig.2. indicates Instruction Decode (ID) stage operation.

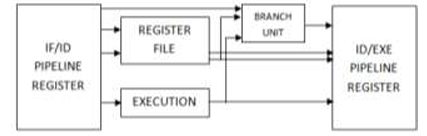


Fig2 Instruction Decoder

1.3 Execute (EX) .

Execution Stage (EX) representation Following the Instruction Decoder(ID), the instructions are sent to execute stage(EXE or EX). Execute (EX) stage performs Arithmetic and Logical Unit (ALU) processes. Execution of operations is the fundamental aspect of Execute (EX) stage, for instance arithmetic operations such as addition and difference and OR & AND. In particular, EX/MEM pipelined register receives the result upon the execution of specific instructions (i.e. FP ALU). Execute stage representation is shown in.

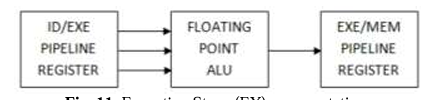


Fig3 Execute

1.4 Memory Access &Input/Output (MEM).

Memory Access representation The storing and loading of values along with inputting and outputting data from the processor is the primary function of the memory access (MEM) stage. The outcome will be dispatched to the WB stage in a scenario where the instruction is neither memory nor IO instruction. After the result is figured the primary function is to store the data values in the destination register. The Memory Access (MEM) stage operation is demonstrated in Figure 4.

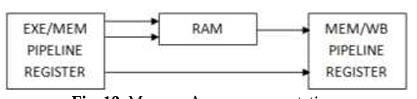


fig4 Memory Access.

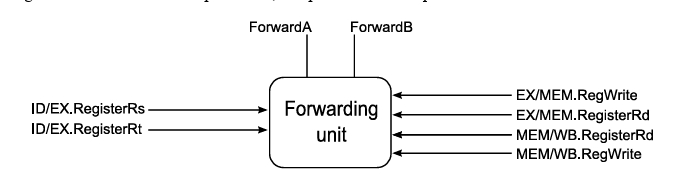
1.5 Write Back (WB) Fig. 13. Write Back representation As per Fig. 5., the Write-Back (WB) operation is the final stage of the RISC based MIPS architecture which composes the result, store information and input data from and to the register . Writing the data that has been fetched from the MIPS register to the target register is the main aim of this stage.



Fig5 Write Back

1.6 The forwarding unit

The forwarding unit takes a total of six input values, and produces two output values: Fig 6: The Forwarding Unit Interface The purpose of the forwarding unit is to guarantee that the instruction entering the EX stage of the pipeline receives the correct values for its register operands.

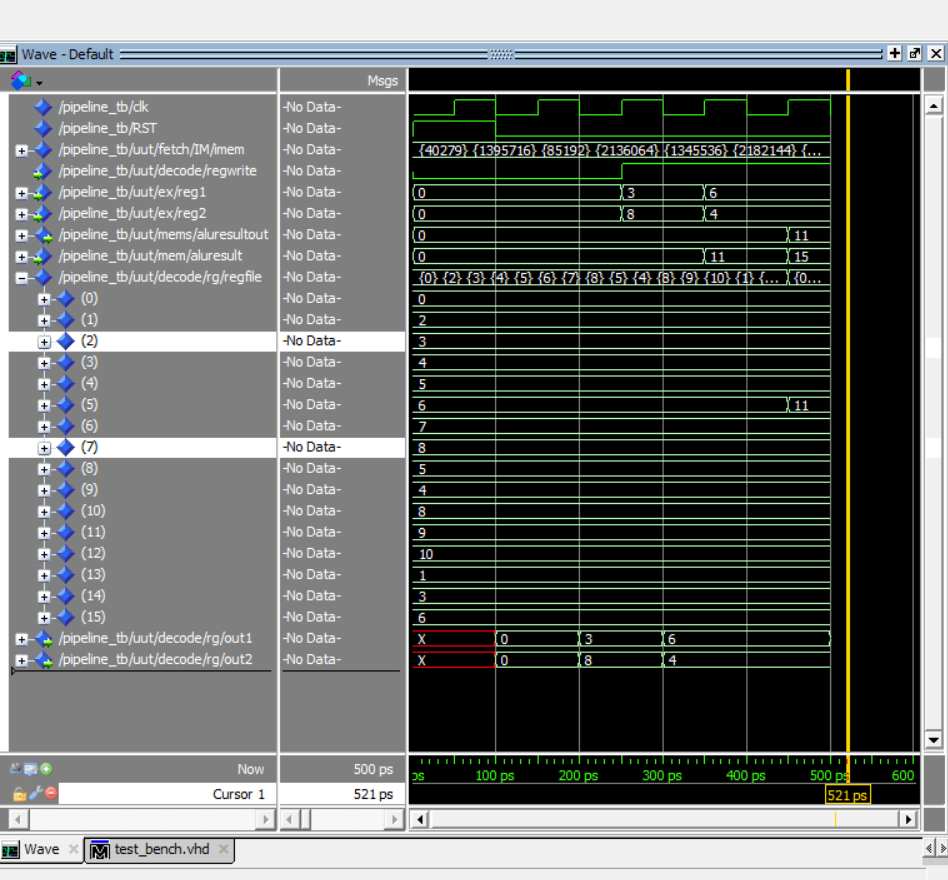
Fig6 forwarding unit

1.6 The hazard detection unit 

The hazard detection unit’s inputs are as follows. — IF/ID.RegisterRs and IF/ID.RegisterRt, the source registers for the current instruction. — ID/EX.MemRead and ID/EX.RegisterRt, to determine if the previous instruction is LW and, if so, which register it will write to.  By inspecting these values, the detection unit generates three outputs. — Two new control signals PCWrite and IF/ID Write, which determine whether the pipeline stalls or continues. — A mux select for a new multiplexer, which forces control signals for the current EX and future MEM/WB stages to 0 in case of a stall.

1.7 Simulation Result

The Resulting output waves of 5-staged pipelined MIPS RISC processor is shown below:



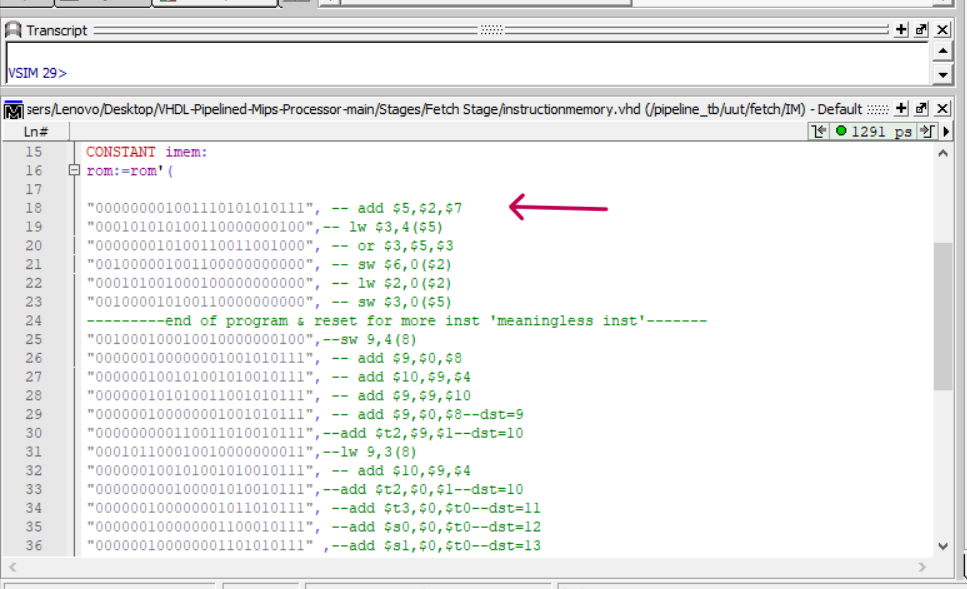
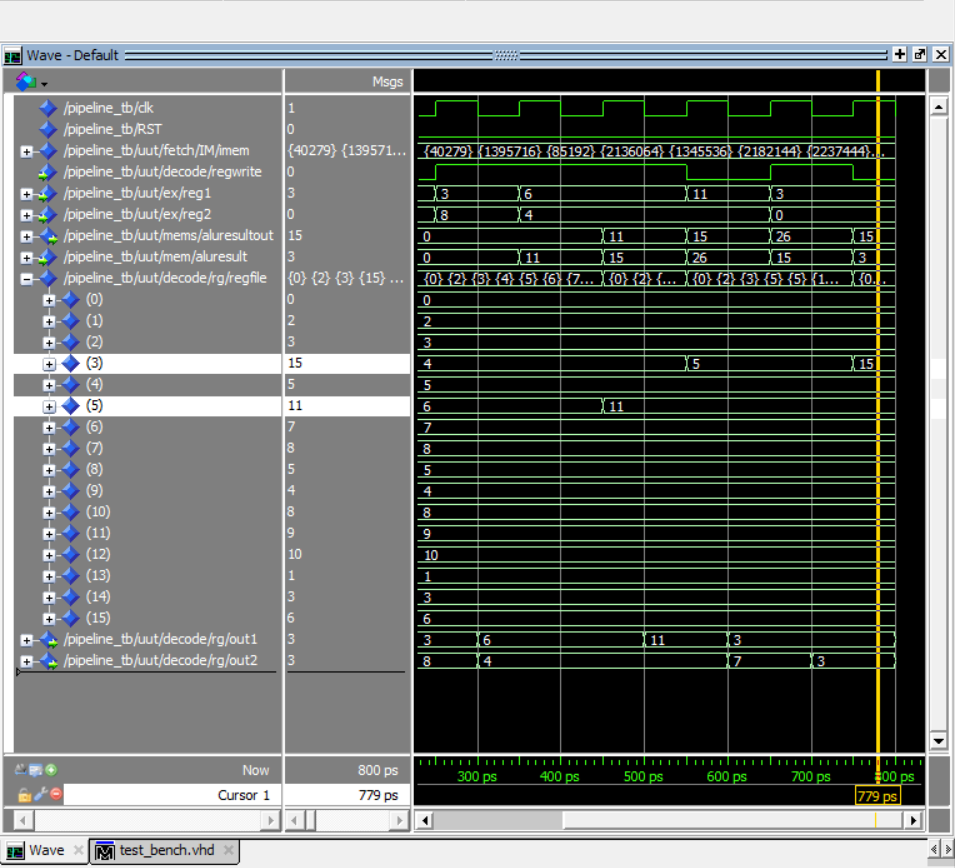


Fig1.code and wave form of instruction 1 (add $5,$2,$7).



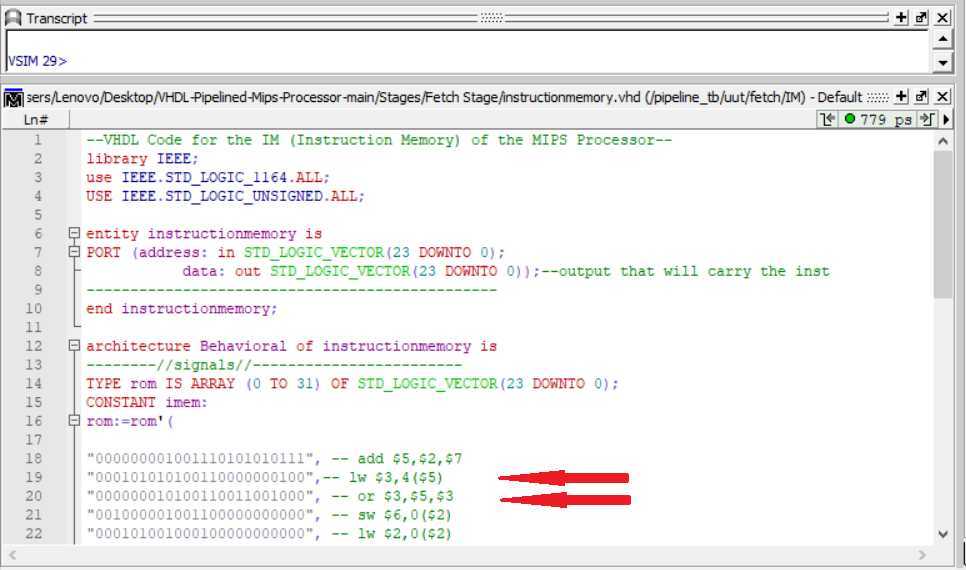


Fig2. assembly code and .wave form of instruction 1,2 (lw $3,4($5)) ------- (or $3,$5,$3).

.

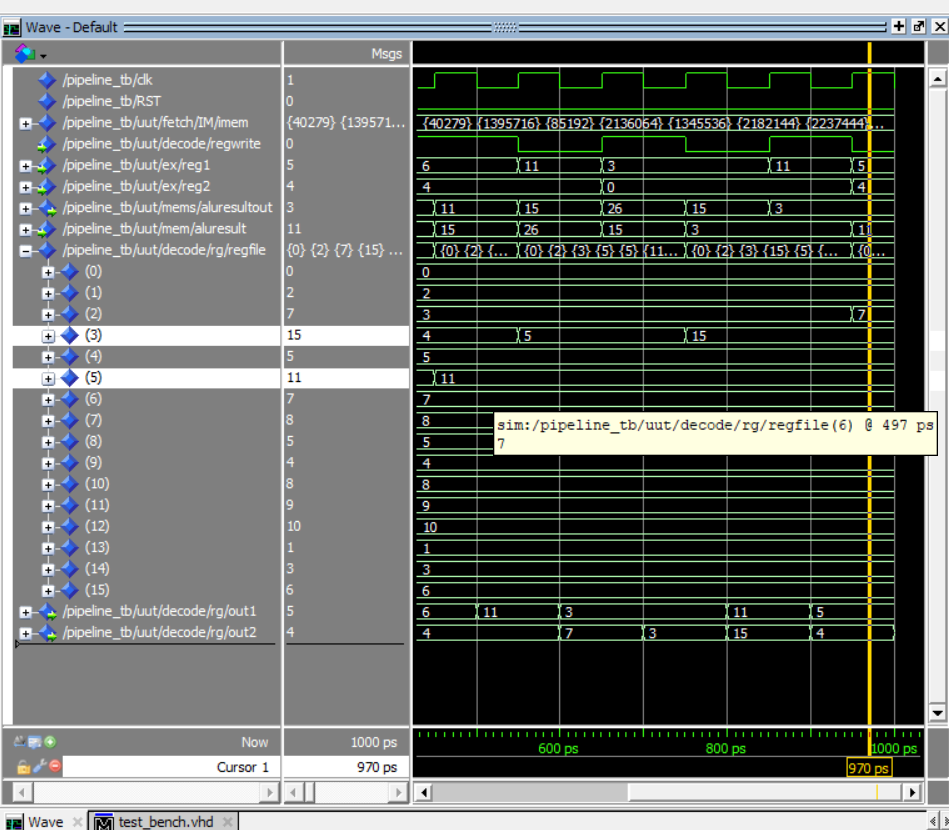


Fig3 .wave form of instruction 4 (sw $6,0($2)).

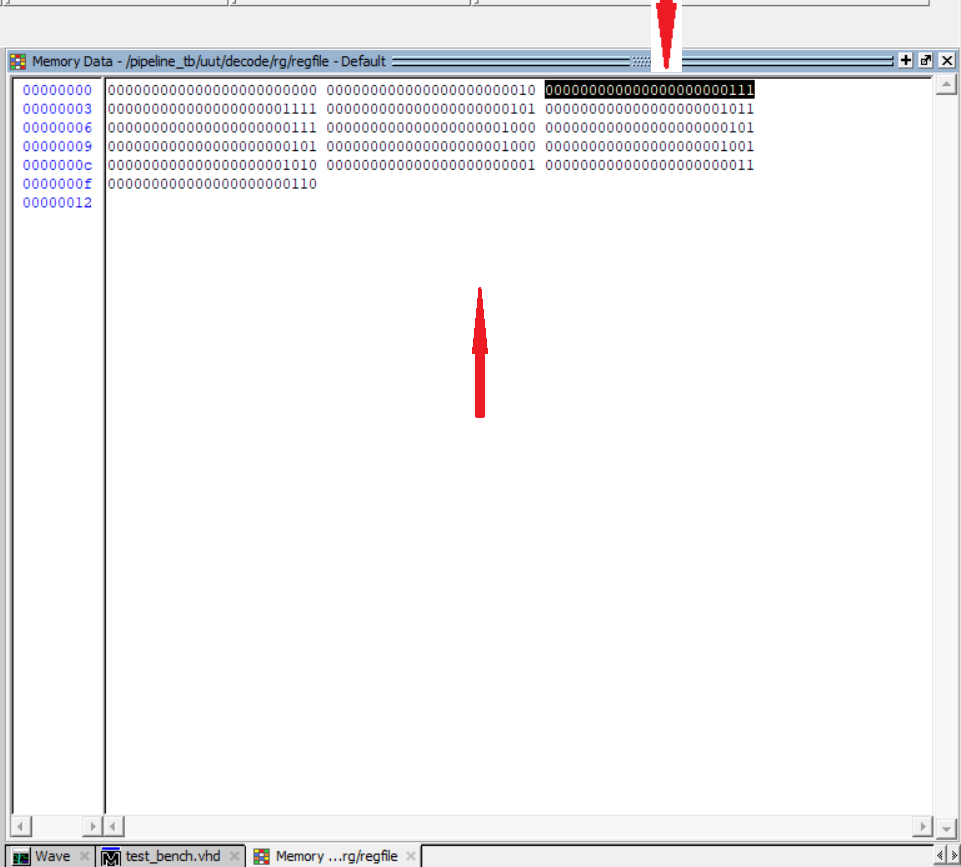


Fig4.data memory content of instruction 4 (sw $6,0($2)).

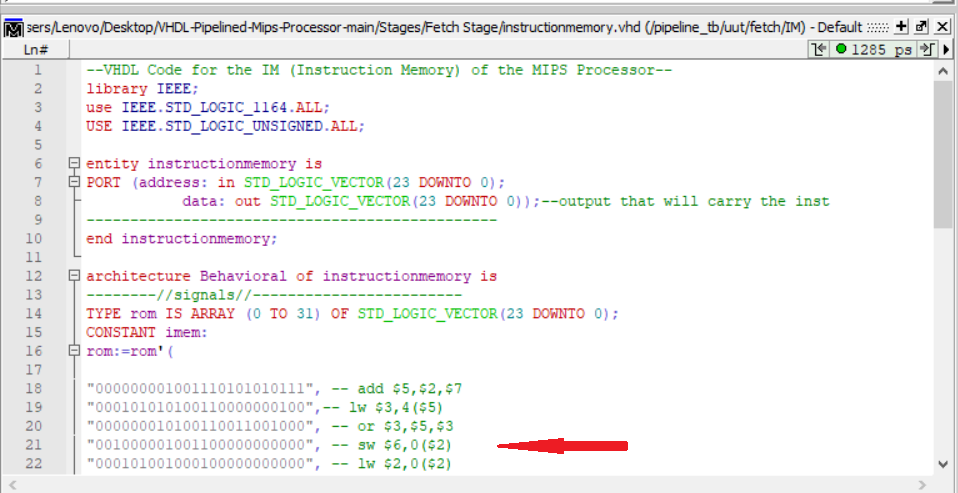


Fig5 .assebly code of intrucrion 4 (sw $6,0($2))

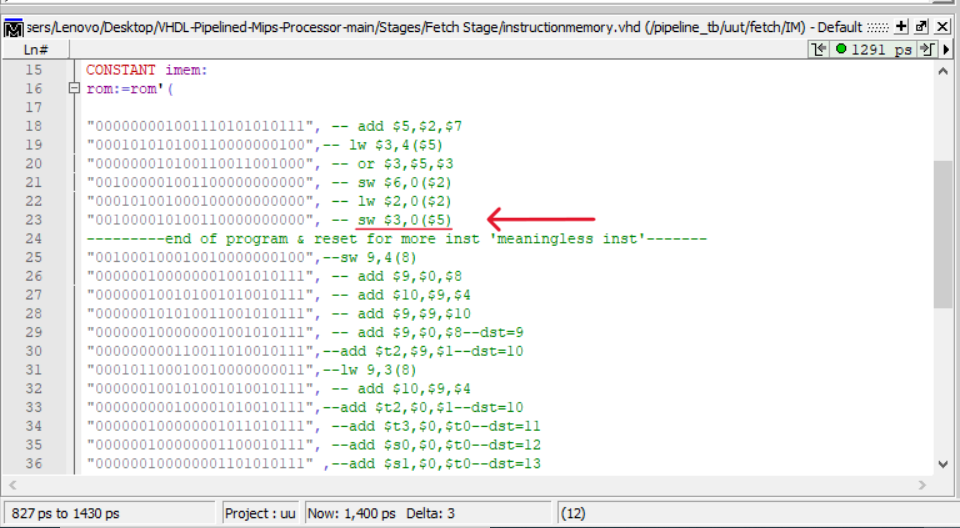


Fig5 .assebly code of intrucrion 5 (sw $3,0($5))

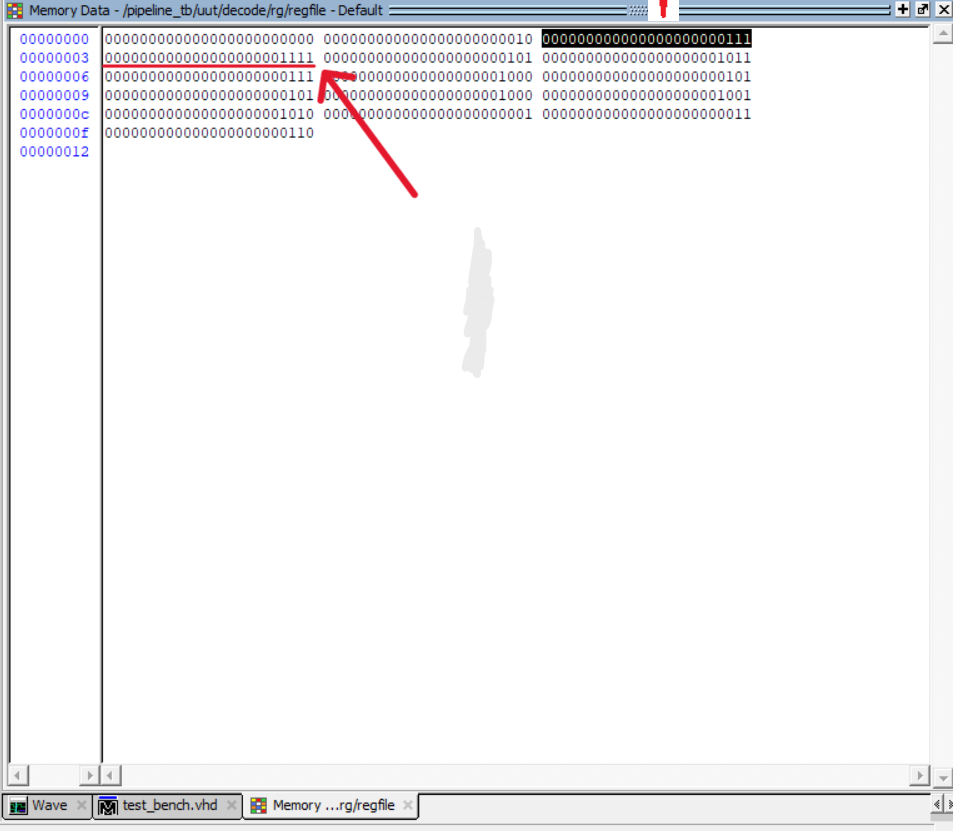


Fig6 .data memory content of instruction 5 (sw $3,0($5)).

1. **Conclusion**

MIPS processor is widely used RISC processor in industry and research area. In this paper, we have successfully designed and synthesized a basic model of pipelined MIPS processor. The design has been modeled in VHDL and functional verification

policies adopted for it. The simulation results show that maximum frequency of pipeline processor is increased from 100MHz to 200MHz.