A person wearing a plaid shirt

Description automatically generated with low confidence

Percobaan IV

SYNTHESIZABLE MIPS32® MICROPROCESSOR  
BAGIAN II : ARITHMETIC AND LOGICAL UNIT (ALU) DAN CONTROL UNIT (CU)

**Ahmad Aziz (13220034)**Asisten: Oktavian Putra Masyiakh (13219039)  
Tanggal Percobaan: 11/11/2022  
EL3111 Praktikum Arsitektur Sistem Komputer  
Laboratorium Sinyal dan Sistem – Sekolah Teknik Elektro dan Informatika  
Institut Teknologi Bandung

*Abstrak*— Pada praktikum modul 4 yaitu SYNTHESIZABLE MIPS32® MICROPROCESSOR BAGIAN II : ARITHMETIC AND LOGICAL UNIT (ALU) DAN CONTROL UNIT (CU), dilakukan beberapa perancangan untuk membuat ALU dan juga CU. Tujuan percobaan pada praktikum modul 4 ini adalah memahami arsitektur mikroprosesor MIPS32® beserta datapath eksekusinya, dapat membuat Arithmetic and Logical Unit (ALU) dari MIPS32® dalam kode VHDL, serta dapat membuat Control Unit (CU) dari MIPS32® dalam kode VHDL. Ada enam tugas yang akan dilakukan pada praktikum modul 4 yaitu Perancangan Program Counter, Perancangan Left Shifter Dua Kali, Perancangan Carry-Lookahead Adder 32-bit, Sign Extender dan Control Unit (CU). Semua percobaan pada praktikum ini dilakukan pada perangkat computer dengan sistem operasi windows dengan software yang digunakan adalah Altera® Quartus® II v9.1sp2 serta code editor menggunakan Visual Studio Code dan juga Notepad++.

*Kata Kunci*—*ALU*, *control unit*, *counter, adder*.

# Pendahuluan

P

ada praktikum modul 4 yaitu SYNTHESIZABLE MIPS32® MICROPROCESSOR

BAGIAN II : ARITHMETIC AND LOGICAL UNIT (ALU) DAN CONTROL UNIT (CU) dilakukan sebanyak 6 percobaan yang bertujuan diantaranya sebagai berikut:

* Praktikan memahami arsitektur mikroprosesor MIPS32® beserta datapath eksekusinya.
* Praktikan dapat membuat Arithmetic and Logical Unit (ALU) dari MIPS32® dalam kode VHDL yang synthesizable dan dapat disimulasikan dengan Altera® Quartus® II v9.1sp2
* Praktikan dapat membuat Control Unit (CU) dari MIPS32® dalam kode VHDL yang synthesizable dan dapat disimulasikan dengan Altera® Quartus® II v9.1sp2.

Praktikum pada modul ini menggunakan software Quartus v9 untuk membuat percobaan arsitektur dan melakukan simulasi. Ada beberapa topik yang dibahas pada modul praktikum ini yaitu sebagai berikut:

1. Perancangan Program Counter
2. Perancangan Left Shifter Dua Kali.
3. Perancangan Carry-Lookahead Adder 32-bit.
4. Sign Extender.
5. Arithmetic and Logical Unit (ALU).
6. Control Unit (CU).

Dalam melakukan percobaan dan analisis pada praktikum modul ini, perangkat lunak dan alat yang digunakan adalah sebagai berikut:

1. Quartus v9
2. PCSpim
3. Code editor Visual Studio Code

# Landasan Teoretis

## Arithmetic and Logical Unit (ALU)

Dalam sistem elektronik digital, sebuah arithmetic and logical unit (ALU) adalah rangkaian digital yang berfungsi untuk melakukan perhitungan integer dan operasi logika. ALU merupakan blok pembangun dasar dari sebuah mikroprosesor. Mikroprosesor modern meliputi central processing unit dan graphics processing unit memiliki ALU yang sangat kompleks untuk melakukan perhitungan. Dalam mikroprosesor modern, digunakan sistem representasi bilangan two’s complement.

Diagram

Description automatically generated

Pada mikroprosesor Single-Cycle MIPS32® yang akan kita realisasikan dalam praktikum ini, terdapat arithmetic and logical unit (ALU) yang sangat sederhana. ALU ini memiliki lebar data input sebesar 32-bit untuk memasukkan dua buah operand dan memiliki lebar data output sebesar 32-bit untuk mengeluarkan hasil komputasi. ALU ini hanya dapat menangani dua operasi matematika saja yaitu penjumlahan dan pengurangan. Untuk operasi penjumlahan, ALU memanfaatkan blok adder. Sedangkan untuk operasi pengurangan, ALU memanfaatkan sifat bilangan two’s complement. Dengan demikian, pengurangan merupakan penjumlahan dengan bilangan negatif. Oleh karena itu, operand kedua dapat diubah menjadi bilangan negatif dengan memanfaatkan prinsip two’s complement yaitu rumus −𝑋 = ~𝑋 + 1. Setelah itu, adder akan menjumlahkan kedua operand tersebut seperti biasa. Untuk memilih operasi penjumlahan dan pengurangan, terdapat 2-to-1 multiplexer yang akan memilih arah operand kedua berasal. Untuk penjumlahan, selektor multiplexer bernilai 0 sedangkan untuk pengurangan selektor multiplexer bernilai 1. Selain itu, carry-in untuk adder juga ditentukan dari operasi yang dilakukan. Untuk penjumlahan, carry-in bernilai 0 sedangkan untuk pengurangan, carry-in untuk bernilai 1. Dengan demikian, kedua sinyal ini (carry-in dan selektor multiplexer) dapat dihubungkan menjadi satu sinyal yaitu OP\_SEL. Untuk melakukan inverting operand kedua, digunakan gerbang NOT dengan lebar data 32-bit.

Untuk mendesain adder, ada beberapa arsitektur adder yang dapat dipilih. Masing-masing arsitektur memiliki kelebihan dan kekurangan yang dapat ditinjau dari segi kecepatan, konsumsi daya, dan konsumsi area. Dua contoh arsitektur adder adalah ripple carry adder dan carry-lookahead adder. Ripple carry adder merupakan adder yang relatif sederhana. Kelemahan adder ini adalah dari segi kecepatan karena setiap bit tidak dapat dijumlahkan secara bersamaan. Tahap adder yang lebih tinggi harus menunggu carry yang dibawa dari tahap adder yang lebih rendah. Pada carry-lookahead adder, setiap tahap adder dapat menghitung carry yang dia terima sehingga tidak perlu menunggu propagasi carry dari tahap sebelumnya. Kelebihan carry-lookahead adder harus dibayar dengan penambahan rangkaian logika yang akan mengkonsumsi luas area.

A picture containing text, sky, clock, sign

Description automatically generated

A picture containing text, clock

Description automatically generated

## Control Unit (CU)

Control Unit (CU) merupakan komponen dari sebuah mikroprosesor yang berfungsi untuk mengarahkan operasioperasi yang dilakukan oleh mikroprosesor tersebut. CU mengatur komunikasi dan koordinasi antarkomponen mikroprosesor menggunakan sinyal-sinyal kontrol. CU juga membaca dan menerjemahkan instruksi-instruksi yang diproses untuk menentukan urutan pemrosesan data.

Diagram, text

Description automatically generated

Pada mikroprosesor Single-Cycle MIPS32® yang akan kita realisasikan dalam praktikum ini, terdapat control unit (CU) yang sangat sederhana. CU menerima opcode dan funct dari instruksi setelah di-decode untuk menentukan nilai dari sinyalsinyal kontrol yang dikeluarkan. Terdapat sepuluh sinyal kontrol yang keluar dari CU ini yang dijelaskan sebagai berikut.

Table

Description automatically generated

Terdapat sembilan instruksi yang dapat dieksekusi oleh mikroprosesor Single-Cycle MIPS32® yang akan kita realisasikan dalam praktikum ini. Kesembilan instruksi tersebut akan menentukan nilai sinyal yang dikeluarkan oleh control unit karena setiap instruksi membutuhkan penanganan dan aliran data yang berbeda-beda. Berikut ini tabel nilai sinyal control unit untuk setiap instruksi yang dapat dieksekusi.

Table

Description automatically generated

Table

Description automatically generated

Untuk mengatur sinyal-sinyal kontrol tersebut, control unit mendeteksi setiap instruksi menggunakan 64 opcode dan funct.

# Hasil dan Analisis

Setelah melakukan percobaan pada semua tugas didapatkan hasil sebagai brerikut:

## Tugas 1: Perancangan Program Counter

Berikut ini adalah hasil simulasi program counter:

Graphical user interface, application, table

Description automatically generated with medium confidence

Timing

Graphical user interface, table

Description automatically generated

## Tugas 2: Perancangan Left Shifter Dua Kali

Berikut ini adalah hasil simulasi program left shift 32 bit:

Function

Graphical user interface, application, table

Description automatically generated

Timing

Graphical user interface, application

Description automatically generated

Dan berikut untuk left shift 26 bit dengan output 28 bit:

Functional

Graphical user interface, application, Word

Description automatically generated

Timing

Graphical user interface, text, application, email

Description automatically generated

## Tugas 3: Perancangan Carry-Lookahead Adder 32-bit

Berikut ini adalah hasil simulasi CLA 32 bit:

Functional

Graphical user interface, application

Description automatically generated

Timing

Graphical user interface

Description automatically generated

## Tugas 4: Sign Extender

## Tugas 5: Arithmetic and Logical Unit (ALU)

## Tugas 6: Control Unit (CU)

# Simpulan

Referensi

1. Bryant, Randal, dan David O’Hallaron. *Computer Systems: A Programmer’s Perspective 2nd Edition*. 2011. Massachusetts: Pearson Education Inc.
2. Patterson, David, dan John Hennessy. *Computer Organization and Design: The Hardware/Software Interface*. 2012. Waltham: Elsevier Inc*.*
3. *Kernighan, Brian, dan Dennis Ritchie. The C Programming Language 2nd edition. 1988. Englewood Cliffs : Prentice Hall.*

**Lampiran**

### Source code untuk tugas 1

program\_counter.vhd

|  |
| --- |
| -- Praktikum EL3111 Arsitektur Sistem Komputer  -- Modul : 4  -- Percobaan : 1  -- Tanggal : 11 November 2022  -- Kelompok : 10  -- Rombongan : B  -- Nama (NIM) 1 : Ahmad Aziz (13220034)  -- Nama (NIM) 2 :  -- Nama File : program\_counter.vhd  -- Deskripsi : Program Counter  **LIBRARY** IEEE**;**  **USE** IEEE**.**STD\_LOGIC\_1164**.ALL;**  **USE** IEEE**.**STD\_LOGIC\_ARITH**.ALL;**  **USE** IEEE**.**STD\_LOGIC\_UNSIGNED**.ALL;**  **LIBRARY** altera\_mf**;**  **USE** altera\_mf**.**altera\_mf\_components**.ALL;**  **ENTITY** program\_counter **IS**  **PORT** **(**  clk **:** **IN** STD\_LOGIC**;**  PC\_in **:** **IN** STD\_LOGIC\_VECTOR **(**31 **DOWNTO** 0**);**  PC\_out **:** **OUT** STD\_LOGIC\_VECTOR **(**31 **DOWNTO** 0**)**  **);**  **END** program\_counter**;**  **ARCHITECTURE** behavior **OF** program\_counter **IS**  **BEGIN**  **PROCESS** **(**clk**,**PC\_in**)**  **BEGIN**  **IF** **(rising\_edge(**clk**))** **THEN**  PC\_out **<=** PC\_in**;**  **END** **IF;**  **END** **PROCESS;**  **END** behavior**;** |

### Source code untuk tugas 2

lshift\_26\_28.vhd

|  |
| --- |
| -- Praktikum EL3111 Arsitektur Sistem Komputer  -- Modul : 4  -- Percobaan : 2b  -- Tanggal : 11 November 2022  -- Kelompok : 10  -- Rombongan : B  -- Nama (NIM) 1 : Ahmad Aziz (13220034)  -- Nama (NIM) 2 :  -- Nama File : lshift\_26\_28.vhd  -- Deskripsi : Program untuk melakukan operasi shift left pada 26 bit dengan output 28 bit  **LIBRARY** IEEE**;**  **USE** IEEE**.**STD\_LOGIC\_1164**.ALL;**  **USE** IEEE**.**STD\_LOGIC\_ARITH**.ALL;**  **USE** IEEE**.**STD\_LOGIC\_UNSIGNED**.ALL;**  **LIBRARY** altera\_mf**;**  **USE** altera\_mf**.**altera\_mf\_components**.ALL;**  **ENTITY** lshift\_26\_28 **IS**  **PORT** **(**  D\_IN **:** **IN** STD\_LOGIC\_VECTOR **(**25 **DOWNTO** 0**);** -- Input 32-bit;  D\_OUT **:** **OUT** STD\_LOGIC\_VECTOR **(**27 **DOWNTO** 0**)** -- Output 32-bit;  **);**  **END** lshift\_26\_28**;**  **ARCHITECTURE** behavior **OF** lshift\_26\_28 **IS**  **BEGIN**  **PROCESS** **(**D\_IN**)**  **BEGIN**  D\_OUT**(**27 **DOWNTO** 2**)** **<=** D\_IN**(**25 **DOWNTO** 0**);**  D\_OUT**(**1 **DOWNTO** 0**)** **<=** "00"**;**  **END** **PROCESS;**  **END** behavior**;** |

lshift\_32\_32.vhd

|  |
| --- |
| -- Praktikum EL3111 Arsitektur Sistem Komputer  -- Modul : 4  -- Percobaan : 2a  -- Tanggal : 11 November 2022  -- Kelompok : 10  -- Rombongan : B  -- Nama (NIM) 1 : Ahmad Aziz (13220034)  -- Nama (NIM) 2 :  -- Nama File : lshift\_32\_32.vhd  -- Deskripsi : Program untuk melakukan operasi shift left 32 bit  **LIBRARY** IEEE**;**  **USE** IEEE**.**STD\_LOGIC\_1164**.ALL;**  **USE** IEEE**.**STD\_LOGIC\_ARITH**.ALL;**  **USE** IEEE**.**STD\_LOGIC\_UNSIGNED**.ALL;**  **LIBRARY** altera\_mf**;**  **USE** altera\_mf**.**altera\_mf\_components**.ALL;**  **ENTITY** lshift\_32\_32 **IS**  **PORT** **(**  D\_IN **:** **IN** STD\_LOGIC\_VECTOR **(**31 **DOWNTO** 0**);** -- Input 32-bit;  D\_OUT **:** **OUT** STD\_LOGIC\_VECTOR **(**31 **DOWNTO** 0**)** -- Output 32-bit;  **);**  **END** lshift\_32\_32**;**  **ARCHITECTURE** behavior **OF** lshift\_32\_32 **IS**  **BEGIN**  **PROCESS** **(**D\_IN**)**  **BEGIN**  D\_OUT**(**31 **DOWNTO** 2**)** **<=** D\_IN**(**29 **DOWNTO** 0**);**  D\_OUT**(**1 **DOWNTO** 0**)** **<=** "00"**;**  **END** **PROCESS;**  **END** behavior**;** |

### Source code untuk tugas 3

cla\_32.vhd

|  |
| --- |
| -- Praktikum EL3111 Arsitektur Sistem Komputer  -- Modul : 4  -- Percobaan : 3  -- Tanggal : 11 November 2022  -- Kelompok : 10  -- Rombongan : B  -- Nama (NIM) 1 : Ahmad Aziz (13220034)  -- Nama (NIM) 2 :  -- Nama File : cla\_32.vhd  -- Deskripsi : Carry Look Ahead Adder 32 bit  **LIBRARY** IEEE**;**  **USE** IEEE**.**STD\_LOGIC\_1164**.ALL;**  **USE** IEEE**.**STD\_LOGIC\_ARITH**.ALL;**  **USE** IEEE**.**STD\_LOGIC\_UNSIGNED**.ALL;**  **LIBRARY** altera\_mf**;**  **USE** altera\_mf**.**altera\_mf\_components**.ALL;**  **ENTITY** cla\_32 **IS**  **PORT** **(**  OPRND\_1 **:** **IN** STD\_LOGIC\_VECTOR**(**31 **DOWNTO** 0**);** -- Operand 1  OPRND\_2 **:** **IN** STD\_LOGIC\_VECTOR **(**31 **DOWNTO** 0**);** -- Operand 2  C\_IN **:** **IN** STD\_LOGIC**;** -- Carry In  RESULT **:** **OUT** STD\_LOGIC\_VECTOR **(**31 **DOWNTO** 0**);** -- Result  C\_OUT **:** **OUT** STD\_LOGIC -- Overflow  **);**  **END** cla\_32**;**  **ARCHITECTURE** behavior **OF** cla\_32 **IS**  **SIGNAL** SUM **:** STD\_LOGIC\_VECTOR**(**31 **DOWNTO** 0**);** -- Sum  **SIGNAL** G **:** STD\_LOGIC\_VECTOR**(**31 **DOWNTO** 0**);** --generate  **SIGNAL** P **:** STD\_LOGIC\_VECTOR**(**31 **DOWNTO** 0**);** --propagate  **SIGNAL** C **:** STD\_LOGIC\_VECTOR**(**31 **DOWNTO** 1**);** --carry    **BEGIN**  SUM **<=** OPRND\_1 **XOR** OPRND\_2**;**  G **<=** OPRND\_1 **AND** OPRND\_2**;**  P **<=** OPRND\_1 **OR** OPRND\_2**;**  **PROCESS** **(**OPRND\_1**,** OPRND\_2**,** C\_IN**)**  **BEGIN**  C**(**1**)** **<=** G**(**0**)** **OR** **(**P**(**0**)** **AND** C\_IN**);**  **FOR** i **IN** 1 **TO** 30 **LOOP**  C**(**i**+**1**)** **<=** G**(**i**)** **OR** **(**P**(**i**)** **AND** C**(**i**));**  **END** **LOOP;**  C\_OUT **<=** G**(**31**)** **OR** **(**P**(**31**)** **AND** C**(**31**));**  RESULT**(**0**)** **<=** SUM**(**0**)** **XOR** C\_IN**;**  RESULT**(**31 **DOWNTO** 1**)** **<=** SUM**(**31 **DOWNTO** 1**)** **XOR** C**(**31 **DOWNTO** 1**);**  **END** **PROCESS;**  **END** behavior**;** |

### Source code untuk tugas 4

reverseByte.h

|  |
| --- |
| -- Praktikum EL3111 Arsitektur Sistem Komputer  -- Modul : 4  -- Percobaan : 4  -- Tanggal : 11 November 2022  -- Kelompok : 10  -- Rombongan : B  -- Nama (NIM) 1 : Ahmad Aziz (13220034)  -- Nama (NIM) 2 :  -- Nama File : sign\_extender.vhd  -- Deskripsi : Program sign extender pada arsitektur MIPS  **LIBRARY** IEEE**;**  **USE** IEEE**.**STD\_LOGIC\_1164**.ALL;**  **USE** IEEE**.**STD\_LOGIC\_ARITH**.ALL;**  **USE** IEEE**.**STD\_LOGIC\_UNSIGNED**.ALL;**  **LIBRARY** altera\_mf**;**  **USE** altera\_mf**.**altera\_mf\_components**.ALL;**  **ENTITY** sign\_extender **IS**  **PORT** **(**  D\_In **:IN** std\_logic\_vector **(**15 **DOWNTO** 0**);** -- Data Input 1  D\_Out **:OUT** std\_logic\_vector **(**31 **DOWNTO** 0**)** -- Data Input 2  **);**  **END** sign\_extender**;**  **ARCHITECTURE** behavior **OF** sign\_extender **IS**  **BEGIN**  **PROCESS** **(**D\_in**)**  **BEGIN**  D\_out**(**31 **DOWNTO** 16**)** **<=** "1111111111111111"**;**  D\_out**(**15 **DOWNTO** 0**)** **<=** D\_IN**(**15 **DOWNTO** 0**);**  **END** **PROCESS;**  **END** behavior**;** |

### Source code untuk tugas 5

ALU.vhd

|  |
| --- |
| -- Praktikum EL3111 Arsitektur Sistem Komputer  -- Modul : 4  -- Percobaan : 5  -- Tanggal : 11 November 2022  -- Kelompok : 10  -- Rombongan : B  -- Nama (NIM) 1 : Ahmad Aziz (13220034)  -- Nama (NIM) 2 :  -- Nama File : ALU.vhd  -- Deskripsi : Program Arithmetic Logic Unit  **LIBRARY** IEEE**;**  **USE** IEEE**.**STD\_LOGIC\_1164**.ALL;**  **USE** IEEE**.**STD\_LOGIC\_ARITH**.ALL;**  **USE** IEEE**.**STD\_LOGIC\_UNSIGNED**.ALL;**  **LIBRARY** altera\_mf**;**  **USE** altera\_mf**.**altera\_mf\_components**.ALL;**  **ENTITY** ALU **IS**  **PORT** **(**  OPRND\_1 **:** **IN** std\_logic\_vector **(**31 **DOWNTO** 0**);** -- Data Input 1  OPRND\_2 **:** **IN** std\_logic\_vector **(**31 **DOWNTO** 0**);** -- Data Input 2  OP\_SEL **:** **IN** std\_logic**;** -- Operation Select  RESULT **:** **OUT** std\_logic\_vector **(**31 **DOWNTO** 0**)** -- Data Output  **);**  **END** ALU**;**  **ARCHITECTURE** behavioral **OF** ALU **IS**  **COMPONENT** cla\_32 **IS**  **PORT** **(**  OPRND\_1 **:** **IN** STD\_LOGIC\_VECTOR**(**31 **DOWNTO** 0**);** -- Operand 1  OPRND\_2 **:** **IN** STD\_LOGIC\_VECTOR **(**31 **DOWNTO** 0**);** -- Operand 2  C\_IN **:** **IN** STD\_LOGIC**;** -- Carry In  RESULT **:** **OUT** STD\_LOGIC\_VECTOR **(**31 **DOWNTO** 0**);** -- Result  C\_OUT **:** **OUT** STD\_LOGIC -- Overflow  **);**  **END** **COMPONENT;**  **SIGNAL** OPRND2\_comp **:** STD\_LOGIC\_VECTOR **(**31 **DOWNTO** 0**);**  **BEGIN** ADDER **:** cla\_32  **PORT** **MAP(**  OPRND\_1 **=>** OPRND\_1**,**  OPRND\_2 **=>** OPRND2\_comp**,**  C\_IN **=>** OP\_SEL**,**  RESULT **=>** RESULT  **);**    **PROCESS** **(**OP\_SEL**,** OPRND\_2**)**  **BEGIN**  **IF** OP\_SEL **=** '0' **THEN**  OPRND2\_comp **<=** OPRND\_2**;**  **ELSE**  OPRND2\_comp **<=** **not** OPRND\_2**;**  **END** **IF;**  **END** **PROCESS;**  **END** behavioral**;** |

### Source code untuk tugas 6

CU.vhd

|  |
| --- |
| -- Praktikum EL3111 Arsitektur Sistem Komputer  -- Modul : 4  -- Percobaan : 6  -- Tanggal : 11 November 2022  -- Kelompok : 10  -- Rombongan : B  -- Nama (NIM) 1 : Ahmad Aziz (13220034)  -- Nama (NIM) 2 :  -- Nama File : CU.vhd  -- Deskripsi : Program control unit (CU)  **LIBRARY** IEEE**;**  **USE** IEEE**.**STD\_LOGIC\_1164**.ALL;**  **USE** IEEE**.**STD\_LOGIC\_ARITH**.ALL;**  **USE** IEEE**.**STD\_LOGIC\_UNSIGNED**.ALL;**  **ENTITY** cu **IS**  **PORT** **(**  OP\_In **:** **IN** STD\_LOGIC\_VECTOR **(**5 **DOWNTO** 0**);**  FUNCT\_In **:** **IN** STD\_LOGIC\_VECTOR **(**5 **DOWNTO** 0**);**  Sig\_Jmp **:** **OUT** STD\_LOGIC**;**  Sig\_Bne **:** **OUT** STD\_LOGIC**;**  Sig\_Branch **:** **OUT** STD\_LOGIC**;**  Sig\_MemtoReg**:** **OUT** STD\_LOGIC**;**  Sig\_MemRead **:** **OUT** STD\_LOGIC**;**  Sig\_MemWrite**:** **OUT** STD\_LOGIC**;**  Sig\_RegDest **:** **OUT** STD\_LOGIC\_VECTOR **(**1 **DOWNTO** 0**);**  Sig\_RegWrite**:** **OUT** STD\_LOGIC**;**  Sig\_ALUSrc **:** **OUT** STD\_LOGIC**;**  Sig\_ALUCtrl **:** **OUT** STD\_LOGIC  **);**  **END** cu**;**  **ARCHITECTURE** behavior **OF** cu **IS**  **SIGNAL** OUT\_sig **:** STD\_LOGIC\_VECTOR**(**10 **DOWNTO** 0**);**  **BEGIN**  **PROCESS** **(**OP\_In**,** FUNCT\_In**)** **IS**  **BEGIN**  **IF** OP\_In **=** "000000" **THEN**  **CASE** FUNCT\_In **IS**  **WHEN** "100000" **=>** OUT\_sig **<=** "00000001100"**;** -- ADD  **WHEN** "100010" **=>** OUT\_sig **<=** "00000001101"**;** -- SUB  **WHEN** **OTHERS** **=>** OUT\_sig **<=** "00000000000"**;** -- NOP  **END** **CASE;**  **ELSE**  **CASE** OP\_In **IS**  **WHEN** "000100" **=>** OUT\_sig **<=** "00100000000"**;** -- BEQ  **WHEN** "000101" **=>** OUT\_sig **<=** "01000000010"**;** -- BNE  **WHEN** "001000" **=>** OUT\_sig **<=** "00000000110"**;** -- ADD  **WHEN** "100011" **=>** OUT\_sig **<=** "00011000110"**;** -- LW  **WHEN** "101011" **=>** OUT\_sig **<=** "00000100010"**;** -- SW  **WHEN** "000010" **=>** OUT\_sig **<=** "10000000000"**;** -- JMP  **WHEN** **others** **=>** OUT\_sig **<=** "00000000000"**;** -- NOP  **END** **CASE;**  **END** **IF;**    Sig\_Jmp **<=** OUT\_sig**(**10**);**  Sig\_Bne **<=** OUT\_sig**(**9**);**  Sig\_Branch **<=** OUT\_sig**(**8**);**  Sig\_MemtoReg **<=** OUT\_sig**(**7**);**  Sig\_MemRead **<=** OUT\_sig**(**6**);**  Sig\_MemWrite **<=** OUT\_sig**(**5**);**  Sig\_RegDest **<=** OUT\_sig**(**4 **DOWNTO** 3**);**  Sig\_RegWrite **<=** OUT\_sig**(**2**);**  Sig\_ALUSrc **<=** OUT\_sig**(**1**);**  Sig\_ALUCtrl **<=** OUT\_sig**(**0**);**  **END** **PROCESS;**  **END** behavior**;** |