Microprocessor 8-bit

Generated by Doxygen 1.6.3

Wed Apr 11 09:49:17 2012

Contents

Chapter 1

Design Unit Index

1.1 Design Unit Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Chapter 2

Design Unit Index

2.1 Design Unit List

Here is a list of all design unit members with links to the Entities and Packages they belong to:

Chapter 3

File Index

3.1 File List

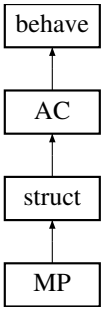
Here is a list of all documented files with brief descriptions:

Chapter 4

Class Documentation

4.1 AC Entity Reference

Inheritance diagram for AC:



Architectures

- behave**Architecture**

Libraries

- ieee

Packages

- std_logic_1164
- std_logic_arith
- std_logic_unsigned

Ports

- d **in std_logic_vector (7 downto 0)**

  8-bit input data to AC from W-bus

- q_alu **out std_logic_vector (7 downto 0)**

  8-bit output data to AC from W-bus

- q_data **out std_logic_vector (7 downto 0)**

  8-bit output data to Adder-Subtractor block

- clk **in std_logic**

  Rising edge clock.

- ea **in std_logic**

  Active high enable AC control input signal.

- clr **in std_logic**

  Active high asynchronous clear.

- la **in std_logic**

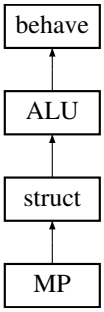
  Active low load AC control input signal.

The documentation for this class was generated from the following file:

- src/ac_behave.vhd

4.2 ALU Entity Reference

Inheritance diagram for ALU:



Architectures

- behave**Architecture**

Libraries

- ieee

Packages

- std_logic_1164
- std_logic_arith
- std_logic_unsigned

Ports

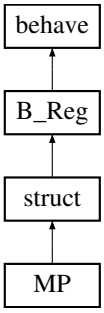
- A **in std_logic_vector (7 downto 0)**
  ALU A input 8-bit from AC.
- B **in std_logic_vector (7 downto 0)**
  ALU B input 8-bit from B-register.
- S **out std_logic_vector (7 downto 0)**
  ALU output 8-bit to W-bus.
- Su **in std_logic**
  Low Add, High Sub.
- Eu **in std_logic**
  Active low enable ALU (tri-state).

The documentation for this class was generated from the following file:

- src/alu_behave.vhd

4.3 B_Reg Entity Reference

Inheritance diagram for B_Reg:



Architectures

- behave**Architecture**

Libraries

- ieee

Packages

- std_logic_1164

Ports

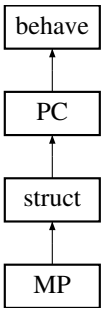
- d **in std_logic_vector (7 downto 0)**
  8-bit B input from W-bus
- q **out std_logic_vector (7 downto 0)**
  8-bit B output to Adder-Subtractor
- clk **in std_logic**
  Rising edge clock.
- clr **in std_logic**
  Active high asynchronous clear.
- lb **in std_logic**
  Active low load B content into output.

The documentation for this class was generated from the following file:

- src/b_reg_behave.vhd

4.4 behave Architecture Reference

Inheritance diagram for behave:



Processes

- PROCESS_7(**clr** , **ep** , **cp** , **clk** , **count**)

Signals

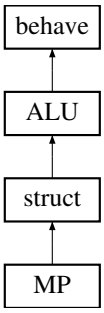
- count **std_logic_vector (3 downto 0)**

The documentation for this class was generated from the following file:

- src/pc_behave.vhd

4.5 behave Architecture Reference

Inheritance diagram for behave:



Processes

- PROCESS_1(**A** , **B** , **Su** , **Eu**)

Signals

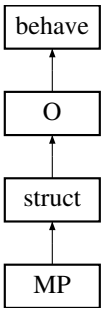
- sub **std_logic_vector (7 downto 0)**
- sum **std_logic_vector (7 downto 0)**

The documentation for this class was generated from the following file:

- src/alu_behave.vhd

4.6 behave Architecture Reference

Inheritance diagram for behave:



Processes

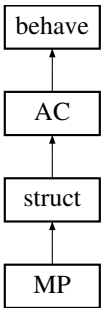
- PROCESS_6(**clr** , **clk** , **lo** , **d**)

The documentation for this class was generated from the following file:

- src/o_behave.vhd

4.7 behave Architecture Reference

Inheritance diagram for behave:



Processes

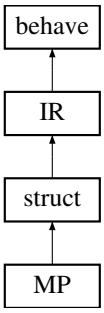
- PROCESS_0(**clr** , **clk** , **la** , **ea** , **d**)

The documentation for this class was generated from the following file:

- src/ac_behave.vhd

4.8 behave Architecture Reference

Inheritance diagram for behave:



Processes

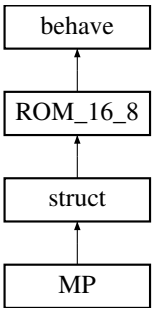
- PROCESS_3(**clr** , **clk** , **li** , **ei**)

The documentation for this class was generated from the following file:

- src/ir_behave.vhd

4.9 behave Architecture Reference

Inheritance diagram for behave:



Processes

- PROCESS_8(**read** , **address**)

Types

- mem **array (0 to 15) of std_logic_vector (7 downto 0)**

Signals

- rom **mem**

4.9.1 Member Function Documentation

4.9.1.1 PROCESS_8(*read* , *address*) [Process]

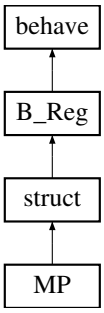
```
This program works as follow:

Load 5 to AC (memory content of 9)
Output 5 (content of AC)
Add 7 (memory content of 10) to 5 (AC content)
Output 12 (content of AC)
Add 3 (memory content of 11) to 12 (AC content)
Subtract 4 (memory content of 12) from 15 (AC content)
Output 11 (content of AC)
```

The documentation for this class was generated from the following file:

- src/rom_16_8_behave.vhd

4.10 behave Architecture Reference

Inheritance diagram for behave:



Processes

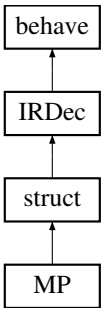
- PROCESS_2(**clr** , **clk** , **lb**)

The documentation for this class was generated from the following file:

- src/b_reg_behave.vhd

4.11 behave Architecture Reference

Inheritance diagram for behave:



Processes

- PROCESS_4(**q_c**)

Signals

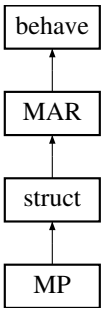
- instruction **std_logic_vector (5 downto 0)**

The documentation for this class was generated from the following file:

- src/irdec_behave.vhd

4.12 behave Architecture Reference

Inheritance diagram for behave:



Processes

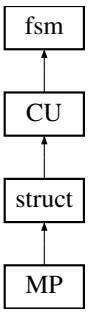
- PROCESS_5(**CLR** , **CLK** , **Lm** , **D**)

The documentation for this class was generated from the following file:

- src/mar_behave.vhd

4.13 CU Entity Reference

Inheritance diagram for CU:



Architectures

- fsm**Architecture**

Libraries

- ieee

Packages

- std_logic_1164
- std_logic_arith
- std_logic_unsigned

Ports

- ADD **in std_logic**

  Add instruction.

- CLK **in std_logic**

  Positive edge trigger clock.

- CLR **in std_logic**

  Active high asynchronous clear.

- LDA **in std_logic**

  Load Accumulator instruction.

- O **in std_logic**

  Out instruction.

- SUB **in std_logic**

  Sub instruction.

- CON **out std_logic_vector (11 downto 0)**

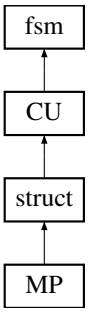
  12-bit control word forming control bus ∼ ∼ ∼ ∼ ∼ ∼ ∼ ∼ CpEpLmCE LiEiLaEa SuEuLbLo

The documentation for this class was generated from the following file:

- src/control_unit_fsm.vhd

4.14 fsm Architecture Reference

Inheritance diagram for fsm:



Processes

- clocked(**CLK** , **CLR**)
- nextstate(**ADD** , **CLR** , **LDA** , **O** , **SUB** , **current_state**)

Types

- STATE_TYPE (**s0** , **s1** , **s2** , **s3** , **s4** , **s5** , **s6** , **s8** , **s9** , **s10** , **s11** , **s12**)

Signals

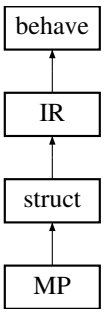
- current_state **STATE_TYPE**
- next_state **STATE_TYPE**

The documentation for this class was generated from the following file:

- src/control_unit_fsm.vhd

4.15 IR Entity Reference

Inheritance diagram for IR:



Architectures

- behave**Architecture**

Libraries

- ieee

Packages

- std_logic_1164

Ports

- clk **in std_logic**

  Rising edge clock.

- clr **in std_logic**

  Active high asynchronous clear.

- li **in std_logic**

  Active low load instruction into IR.

- ei **in std_logic**

  Active low enable IR output.

- d **in std_logic_vector (7 downto 0)**

  IR 8-bit input data word from W-bus.

- q_w **out std_logic_vector (3 downto 0)**

  IR 4-bit output data word to W-bus.

- q_c **out std_logic_vector (3 downto 0)**

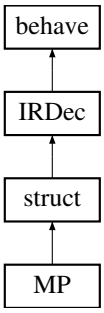
IR 4-bit output control word to Control-Sequencer block.

The documentation for this class was generated from the following file:

- src/ir_behave.vhd

4.16 IRDec Entity Reference

Inheritance diagram for IRDec:



Architectures

- behave**Architecture**

Libraries

- ieee

Packages

- std_logic_1164

Ports

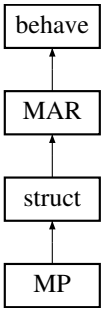
- q_c **in std_logic_vector (3 downto 0)**
- LDA **out std_logic**
- ADD **out std_logic**
- SUB **out std_logic**
- OUTPUT **out std_logic**
- HLT **out std_logic**

The documentation for this class was generated from the following file:

- src/irdec_behave.vhd

4.17 MAR Entity Reference

Inheritance diagram for MAR:



Architectures

- behave**Architecture**

Libraries

- ieee

Packages

- std_logic_1164
- std_logic_arith
- std_logic_unsigned

Ports

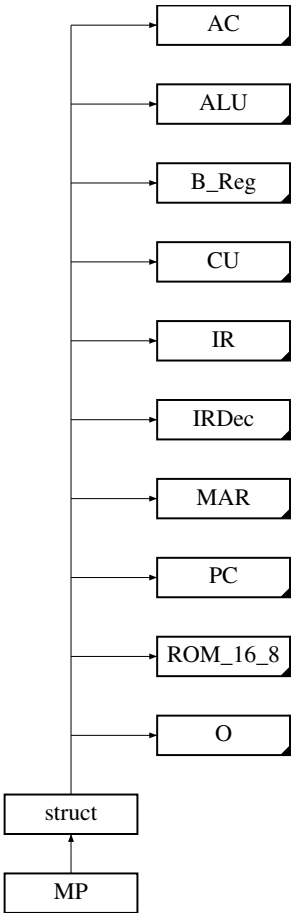
- CLK **in std_logic**
  Rising edge clock.
- CLR **in std_logic**
  Active high asynchronous clear.
- Lm **in std_logic**
  Active low load MAR.
- D **in std_logic_vector (3 downto 0)**
  MAR 4-bit address input.
- Q **out std_logic_vector (3 downto 0)**
  MAR 4-bit address output.

The documentation for this class was generated from the following file:

- src/mar_behave.vhd

4.18 MP Entity Reference

Inheritance diagram for MP:



Architectures

- struct**Architecture**

Libraries

- ieee

Packages

- std_logic_1164
- std_logic_arith

Ports

- clk **in std_logic**

Active high asynchronous clear.

- clr **in std_logic**

  Rising edge clock.

- hlt **out std_logic**

  Halt signal to stop processing data.

- q3 **out std_logic_vector (7 downto 0)**

  8-bit output

4.18.1 Member Data Documentation

4.18.1.1 ieee library `[Library]`

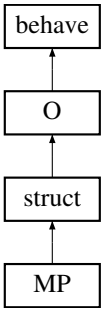
Reimplemented from AC.

The documentation for this class was generated from the following file:

- src/MP_struct.vhd

4.19 O Entity Reference

Inheritance diagram for O:



Architectures

- behave**Architecture**

Libraries

- ieee

Packages

- std_logic_1164

Ports

- d **in std_logic_vector (7 downto 0)**

  8-bit O input from W-bus

- q **out std_logic_vector (7 downto 0)**

  8-bit O output

- clk **in std_logic**

  Rising edge clock.

- clr **in std_logic**

  Active high asynchronous clear.

- lo **in std_logic**

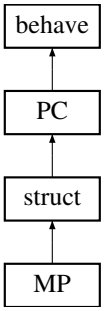
  Active low load O content into output.

The documentation for this class was generated from the following file:

- src/o_behave.vhd

4.20 PC Entity Reference

Inheritance diagram for PC:



Architectures

- behave**Architecture**

Libraries

- ieee

Packages

- std_logic_1164
- std_logic_unsigned

Ports

- ep **in std_logic**

  Active high otuput enable from PC, or tri-state.

- clr **in std_logic**

  Active high asynchronous clear.

- clk **in std_logic**

  Falling edge clock.

- cp **in std_logic**

  Active high enable PC to count.

- q **out std_logic_vector (3 downto 0)**

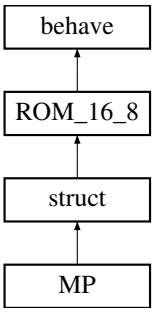
  4-bit PC output

The documentation for this class was generated from the following file:

- src/pc_behave.vhd

4.21 ROM_16_8 Entity Reference

Inheritance diagram for ROM_16_8:



Architectures

- behave**Architecture**

Libraries

- ieee

Packages

- std_logic_1164
- std_logic_arith
- std_logic_unsigned

Ports

- read **in std_logic**

  Active low enable ROM signal, (tri-state).

- address **in std_logic_vector (3 downto 0)**

  4-bit ROM address bits from MAR

- data_out **out std_logic_vector (7 downto 0)**

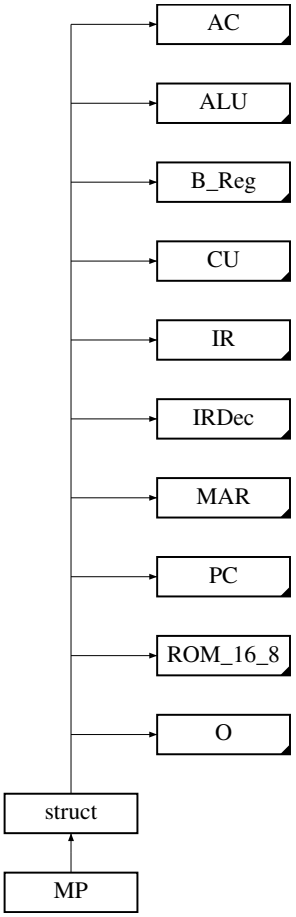
  8-bit ROM output word to W-bus

The documentation for this class was generated from the following file:

- src/rom_16_8_behave.vhd

4.22 struct Architecture Reference

Inheritance diagram for struct:



Components

- AC
- ALU
- B_Reg
- CU
- IR
- IRDec
- MAR
- PC
- ROM_16_8
- O

Signals

- Ce **std_logic**

Chip select for ROM.

- D **std_logic_vector (3 DOWNTO 0)**

  MAR 4-bit address input.

- Eu **std_logic**

  Enable ALU.

- Lm **std_logic**

  Content of PC are latched into MAR on the next +ve edge (LOW).

- Q2 **std_logic_vector (3 DOWNTO 0)**

  MAR 4-bit address output.

- Su **std_logic**

  Add or Sub.

- W **std_logic_vector (7 DOWNTO 0)**

  W-bus the major internal data bus.

- add **std_logic**

  IR decoder add control signal.

- con **std_logic_vector (11 DOWNTO 0)**

  Control word bus.

- Cp **std_logic**

  Chip select PC.

- d1 **std_logic_vector (7 DOWNTO 0)**

  8-bit output data to Adder-Subtractor block

- Ea **std_logic**

  Enable AC.

- Ei **std_logic**

  Enable IR.

- Ep **std_logic**

  Enable PC.

- La **std_logic**

  Load Accumulator AC.

- Lb **std_logic**

  Load B Register B.

- lda **std_logic**

  Load Accumulator instruction.

- Li **std_logic**

  Load Instruction Register IR.

- Lo **std_logic**

  Load Output Register O.

- output **std_logic**

  Output the result.

- q **std_logic_vector (3 DOWNTO 0)**

  4-bit PC output

- q1 **std_logic_vector (7 DOWNTO 0)**

  ALU B input 8-bit from B-register.

- q_alu **std_logic_vector (7 DOWNTO 0)**

  ALU A input 8-bit from AC.

- q_c **std_logic_vector (3 DOWNTO 0)**

  IR 4-bit output control word to Control-Sequencer block.

- q_w **std_logic_vector (3 DOWNTO 0)**

  IR 4-bit output data word to W-bus.

- sub **std_logic**

  IR decoder sub control signal.

Component Instantiations

- Accumulator**AC**
- AddSub**ALU**
- BReg**B_Reg**
- CPU**CU**
- IRReg**IR**
- IRDecoder**IRDec**
- MemoryAddressReg**MAR**
- ProgramCounter**PC**
- ROM**ROM_16_8**
- OReg**O**

The documentation for this class was generated from the following file:

- src/MP_struct.vhd

Chapter 5

File Documentation

5.1 src/ac_behave.vhd File Reference

Accumulator (AC).

Architectures

- AC**Entity**
- behave**Architecture**

5.1.1 Detailed Description

Accumulator (AC). is a buffer register that stores intermediate amswers during a computer run. It is connected directly to the W-bus (3-state) and Adder-Subtractor/ALU (2-state).

5.2 src/alu_behave.vhd File Reference

Arithmetic Logic Unit (ALU).

Architectures

- ALU**Entity**
- behave**Architecture**

5.2.1 Detailed Description

Arithmetic Logic Unit (ALU). It just perform addition and subtraction operation.

It is asynchronous block.

5.3 src/b_reg_behave.vhd File Reference

B register (B).

Architectures

- B_Reg**Entity**
- behave**Architecture**

5.3.1 Detailed Description

B register (B). It is another buffer register. It is used in arithmetic operations.

Its input connected to the W-bus, it transfer the data in when Lb is low.

Its output connected to ALU B input.

5.4 src/control_unit_fsm.vhd File Reference

Controller-Sequencer (CU).

Architectures

- CU**Entity**
- fsm**Architecture**

5.4.1 Detailed Description

Controller-Sequencer (CU). The output is 12-bit form a word controlling the rest of the processor.

It is called the contol bus.

CON = Cp Ep nLm nCE nLi nEi nLa Ea Su Eu nLb nLo

The control word determines how the registers will react to the next clock edge.

P.S. n for activ low signal

5.5 src/ir_behave.vhd File Reference

Instruction Register (IR).

Architectures

- IR**Entity**
- behave**Architecture**

5.5.1 Detailed Description

Instruction Register (IR). It is a part of the control unit.

The output of the IR is 8-bit word. It is divided into two nibbles.

Upper Nibble Lower Nibble

2-state 3-state

CU W-bus

The provided instruction set is:

LDA 0000 Load Accumulator with corresponding memory content

ADD 0001 Add the content of the AC to the content of the memory adder

SUB 0010 Subtract the content of the memory location from the AC

OUT 1110 Transfer the AC content to the output port

HLT 1111 Stop processing data

Fetch = 3 cycles Execute = 3 cycles

5.6 src/irdec_behave.vhd File Reference

Instruction Register Decoder (IRDec).

Architectures

- IRDec**Entity**
- behave**Architecture**

5.6.1 Detailed Description

Instruction Register Decoder (IRDec). It is equivelent to a ring counter driving the CU.

5.7 src/mar_behave.vhd File Reference

Memory Address Register (MAR).

Architectures

- MAR**Entity**
- behave**Architecture**

5.7.1 Detailed Description

Memory Address Register (MAR). It is part of the processor memory. During a computer run, the address in the PC is latched into the MAR.

A bit later, the MAR applies this 4-bit address to the RAM, where a read operation is performed.

5.8 src/MP_struct.vhd File Reference

This is the top-level design for a simple 8-bit microprossesor.

Architectures

- MP**Entity**
- struct**Architecture**

5.8.1 Detailed Description

This is the top-level design for a simple 8-bit microprossesor. This is a 8-bit microprocessor which is know as SAP-1 or Simple-As-Possible Computer. It is described in [1].

Author

Ahmed Shahein ahmed.shahein@ieee.org

See also

[1] Malvino, A.P. and Brown, J.A., "Digital computer electronics", Glencoe/McGraw-Hill, 1992.

5.9 src/o_behave.vhd File Reference

Output Register (O).

Architectures

- O**Entity**
- behave**Architecture**

5.9.1 Detailed Description

Output Register (O). This buffer is used to transfer the answer to the probelm being solved to the outside world.

At high Ea and low Lo at next clock edge the content of the AC is loaded into the O register.

5.10 src/pc_behave.vhd File Reference

Program Counter (PC).

Architectures

- PC**Entity**
- behave**Architecture**

5.10.1 Detailed Description

Program Counter (PC). The PC is reset to 0000 before the processor runs. Then the PC send the address 0000 to the RAM/ROM,

to fetch and exectute the corresponding instruction. After the first instruction is fetched and exectuted

the PC sends the following address 0001 to the RAM/ROM, and so on.

The PC is part of the conrtol unit, it counts from 0000 to 1111.

It is called pointer; it points to a memory location where instruction is stored.

It work as 4-bit counter.

5.11 src/rom_16_8_behave.vhd File Reference

Read Only Memory.

Architectures

- ROM_16_8**Entity**
- behave**Architecture**

5.11.1 Detailed Description

Read Only Memory. It is used to store the program on it. It replaces a RAM on the original design.

Index