

APPENDIX C

VHDL DESCRIPTION AND SYNTHESIS OF MIPS R2000 CONTROL UNIT

This appendix presents the development of the Register Transfer Level (RTL) description of the control unit of the MIPS R2000 microprocessor. The control unit concepts are first reviewed and then followed by my own work on implementing this description in VHDL. This VHDL description (also called RTL Model) of this control unit includes simulation and synthesis onto the target Xilinx Virtex-II FPGA chip. Again, this appendix is based on and complements the material presented in [47] and [48] and is annotated with my comments and tailored adaptation for the context of this research. This appendix is the last piece of the big picture used to create the VHDL description and synthesis of the finalized full MIPS R2000 microprocessor in chapter 6.

C.1 Introduction

This appendix presents the VHDL description, synthesis and simulation of the MIPS R2000 control unit.

The format for presentation of the material in this appendix is the same as that in Appendices A and B, where I take each unit and briefly review its RTL description as described in [47] and [48], then follow it with my own work implementing this unit in VHDL, along with its synthesis and simulation. This process follows the design cycle (described in chapter 4) and comprised of the following steps: RTL Description, Design Entry and Synthesis, Synthesis Results, FPGA Device Synthesis Summary, Place-and-Route onto the FPGA, and Simulation Results. Also, the logic conventions and clocking methodology followed in this appendix are detailed in Appendix A (section A.2).

This appendix starts by delving right into the details of the Control Unit in section C.2, which covers the ALU Control first then the Main Control. Section C.3 concludes this appendix with a summary.

C.2 The Control Unit

Now that the complete datapath for MIPS R2000 was finalized in the previous appendix, the Control Unit can now be designed. This control unit accepts inputs from the instruction and generates the necessary read/write signals for each state element, the selector control for each multiplexer, and the ALU Control signals [47]. Since the ALU Control is different in a number of ways, it is best to design it first before designing the rest of the control unit [47].

C.2.1 ALU Control Unit

➤ RTL Description

The ALU Control Unit is discussed in detail on pages 353 to 356 and pages C-4 to C-7 of [47]. The design and hardware implementation of the ALU control unit for the context of this research is exactly the same as that outlined in [47]. However, Figure C.1 outlines in more detail the functionality and derivation of the control bits needed for the ALU Control Unit. Figure C.2 shows the structured gate-level implementation of finalized ALU control unit derived from the table in Figure C.1.

Instruction	Opcode Field	Funct Field	ALU Function	ALU Operation		ALUOp
				Binary	Decimal	
AND	0	36	AND	000	0	10
OR	0	37	OR	001	1	10
ADD	0	32	ADD	010	2	10
SUB	0	34	SUB	110	6	10
SLT	0	42	SLT	111	7	10
LW	35	–	ADD	010	2	00 (add)
SW	43	–	ADD	010	2	00 (add)
J	2	–	–	–	–	–
BEQ	4	–	SUB	110	6	01 (sub)
BNE	5	–	SUB	110	6	01 (sub)

Figure C.1 How the ALU Control Unit bits are set (adapted from [47]).

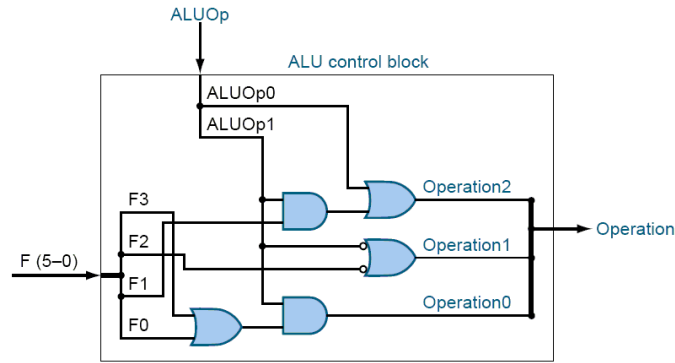


Figure C.2 Structured gate-level implementation of the ALU Control Unit [47, p.C-6].

➤ Design Entry and Synthesis

Schematic Editor was used to create the design entry for the ALU Control Unit shown in figure C.1. Figure C.3 shows the final schematic diagram.

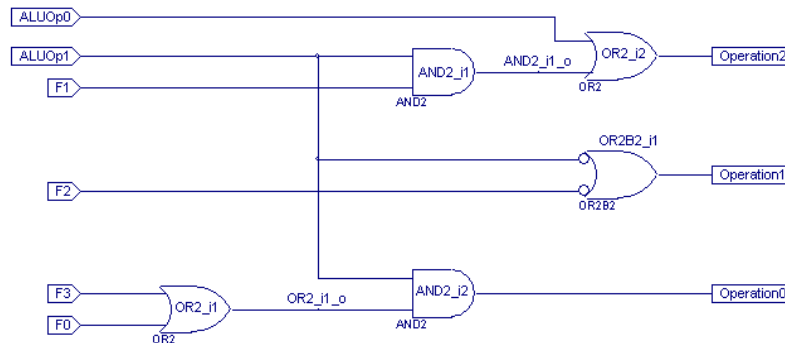


Figure C.3 Schematic diagram design entry in Schematic Editor for the ALU Control Unit.

After synthesis of the schematic diagram in figure C.2 using XST, the following VHDL code was generated:

```
-- Vhdl model created from schematic alu_control.sch - Fri May 14 12:16:11 2004

LIBRARY ieee;
USE ieee.std_logic_1164.ALL;
USE ieee.numeric_std.ALL;
-- synopsys translate_off
LIBRARY UNISIM;
USE UNISIM.Vcomponents.ALL;
-- synopsys translate_on

ENTITY alu_control IS
    PORT (
        ALUOp0      : IN    STD_LOGIC;
        ALUOp1      : IN    STD_LOGIC;
        F0          : IN    STD_LOGIC;
        F1          : IN    STD_LOGIC;
        F2          : IN    STD_LOGIC;
        F3          : IN    STD_LOGIC;
        Operation0   : OUT   STD_LOGIC;
        Operation1   : OUT   STD_LOGIC;
        Operation2   : OUT   STD_LOGIC);
```

```

end alu_control;

ARCHITECTURE SCHEMATIC OF alu_control IS
    SIGNAL AND2_i1_o      :      STD_LOGIC;
    SIGNAL OR2_i1_o       :      STD_LOGIC;

    ATTRIBUTE BOX_TYPE : STRING;

    COMPONENT AND2
        PORT ( I0      :      IN      STD_LOGIC;
              I1      :      IN      STD_LOGIC;
              O        :      OUT     STD_LOGIC);
    END COMPONENT;

    ATTRIBUTE BOX_TYPE OF AND2 : COMPONENT IS "BLACK_BOX";
    COMPONENT OR2
        PORT ( I0      :      IN      STD_LOGIC;
              I1      :      IN      STD_LOGIC;
              O        :      OUT     STD_LOGIC);
    END COMPONENT;

    ATTRIBUTE BOX_TYPE OF OR2 : COMPONENT IS "BLACK_BOX";
    COMPONENT OR2B2
        PORT ( I0      :      IN      STD_LOGIC;
              I1      :      IN      STD_LOGIC;
              O        :      OUT     STD_LOGIC);
    END COMPONENT;

    ATTRIBUTE BOX_TYPE OF OR2B2 : COMPONENT IS "BLACK_BOX";
BEGIN

    AND2_i1 : AND2
        PORT MAP (I0=>F1, I1=>ALUOp1, O=>AND2_i1_o);

    AND2_i2 : AND2
        PORT MAP (I0=>OR2_i1_o, I1=>ALUOp1, O=>Operation0);

    OR2_i2 : OR2
        PORT MAP (I0=>AND2_i1_o, I1=>ALUOp0, O=>Operation2);

    OR2_i1 : OR2
        PORT MAP (I0=>F0, I1=>F3, O=>OR2_i1_o);

    OR2B2_i1 : OR2B2
        PORT MAP (I0=>F2, I1=>ALUOp1, O=>Operation1);

END SCHEMATIC;

```

➤ *Synthesis Results*

Using the Xilinx ISE synthesis tools, the hardware implementation for the ALU Control Unit, was generated. Figure C.4 shows the resulting top level RTL symbol while figure C.5 shows the resulting top level RTL schematic diagram.

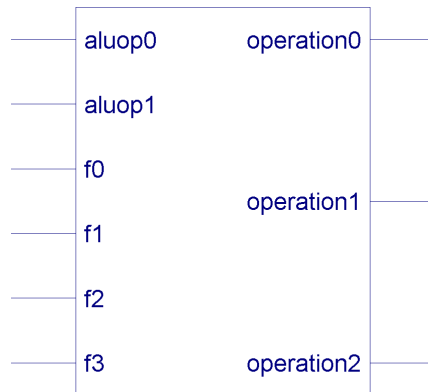


Figure C.4 Resulting top level RTL symbol for the ALU Control Unit.

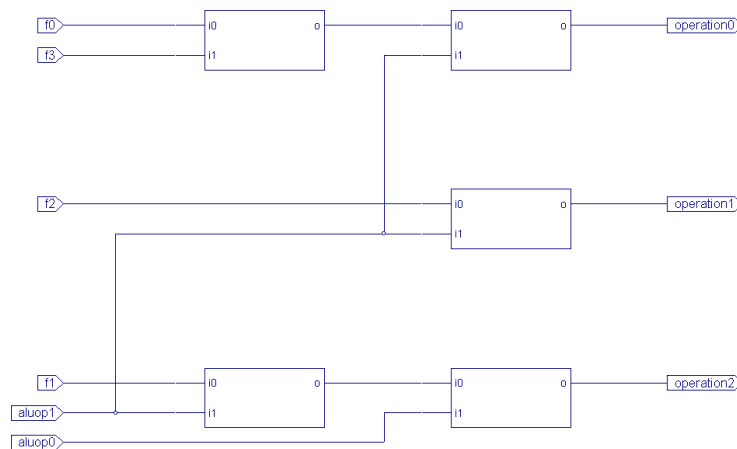


Figure C.5 Resulting top level RTL schematic for the ALU Control Unit.

➤ FPGA Device Synthesis Summary

After the hardware implementation for this ALU Control Unit using the Xilinx ISE synthesis tools, the Synthesis Report was generated. The most important FPGA Device Synthesis Statistics from this report, are shown below:

Design Statistics:

IOs : 9

Cell Usage:

BELS : 5
 # and2 : 2
 # or2 : 2
 # or2b2 : 1
 # IO Buffers : 9
 # IBUF : 6
 # OBUF : 3

Device utilization summary:

Number of bonded IOBs: 9 out of 1108 0%

➤ *Place-and-Route onto the FPGA*

In figure C.6, FPGA Editor shows the synthesized ALU Control Unit after place-and-route onto the target Virtex-II FPGA chip. Notice that these are the blue interconnections running across the very top of the FPGA chip.

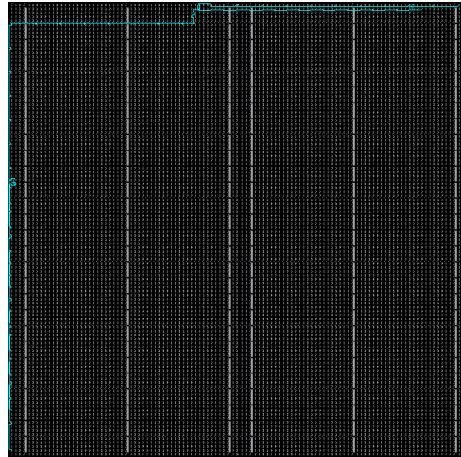


Figure C.6 *FPGA Editor showing the synthesized ALU Control Unit after place-and-route onto the target Virtex-II FPGA chip.*

➤ *Simulation Results*

Figure C.7 shows the waveform results of simulating the ALU Control Unit VHDL behavioural model in Mentor Graphics ModelSim. These waveforms are in binary format. It is clear that the resulting synthesized hardware functions according to the specified behavior of the ALU Control Unit. This concludes the design cycle for this component.

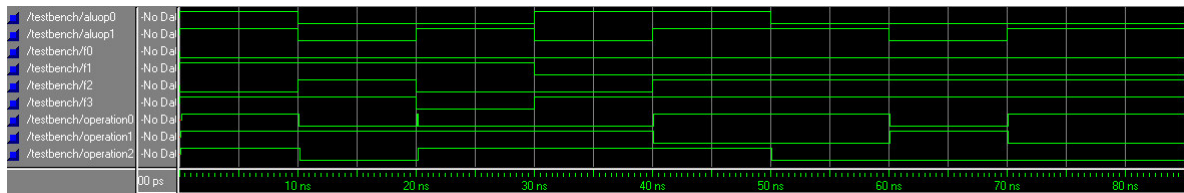


Figure C.7 *Results of simulating the synthesized ALU Control Unit, using ModelSim.*

C.2.2 Main Control Unit

➤ *RTL Description*

The Main Control Unit is discussed in detail on pages 356 to 370 and pages C-7 to C-8 of [47]. However, the design and hardware implementation of the Main Control Unit for the context of this research slightly deviates from that outlined in [47]. This is attributed to the extra multiplexers and design modifications introduced earlier on in Appendices A and B. Figure C.8 outlines the truth table for the finalized Main Control Unit. Figure C.9 shows the structured gate-level implementation of the finalized Main control unit derived from the truth table in Figure C.8.

Input Or Output	Signal Name	R-format	LW	SW	BEQ	BNE	J
Inputs	Op5	0	1	1	0	0	0
	Op4	0	0	0	0	0	0
	Op3	0	0	1	0	0	0
	Op2	0	0	0	1	1	0
	Op1	0	1	1	0	0	1
	Op0	0	1	1	0	1	0
Outputs	ALUOp1	1	0	0	0	0	X
	ALUOp0	0	0	0	1	1	X
	MemtoReg1	0	0	X	X	X	X
	MemtoReg0	0	1	X	X	X	X
	RegDst1	1	0	X	X	X	X
	RegDst0	0	1	X	X	X	X
	ALUMuxEn	1	1	1	1	1	0
	Branch	0	0	0	1	1	0
	MemRead	0	1	0	0	0	0
	MemWrite	0	0	1	0	0	0
	SelBEQorBNE	X	X	X	0	1	X
	Jump	0	0	0	0	0	1
	RegRW	1	1	1	1	1	0
	ALUSrc	0	1	1	0	0	X
	DMemDinSrc	X	X	1	X	X	X
	DMemRASrc	X	1	X	X	X	X
	DmemWASrc	X	X	1	X	X	X

Figure C.8 Truth table for the finalized Main Control Unit (adapted from [47]).

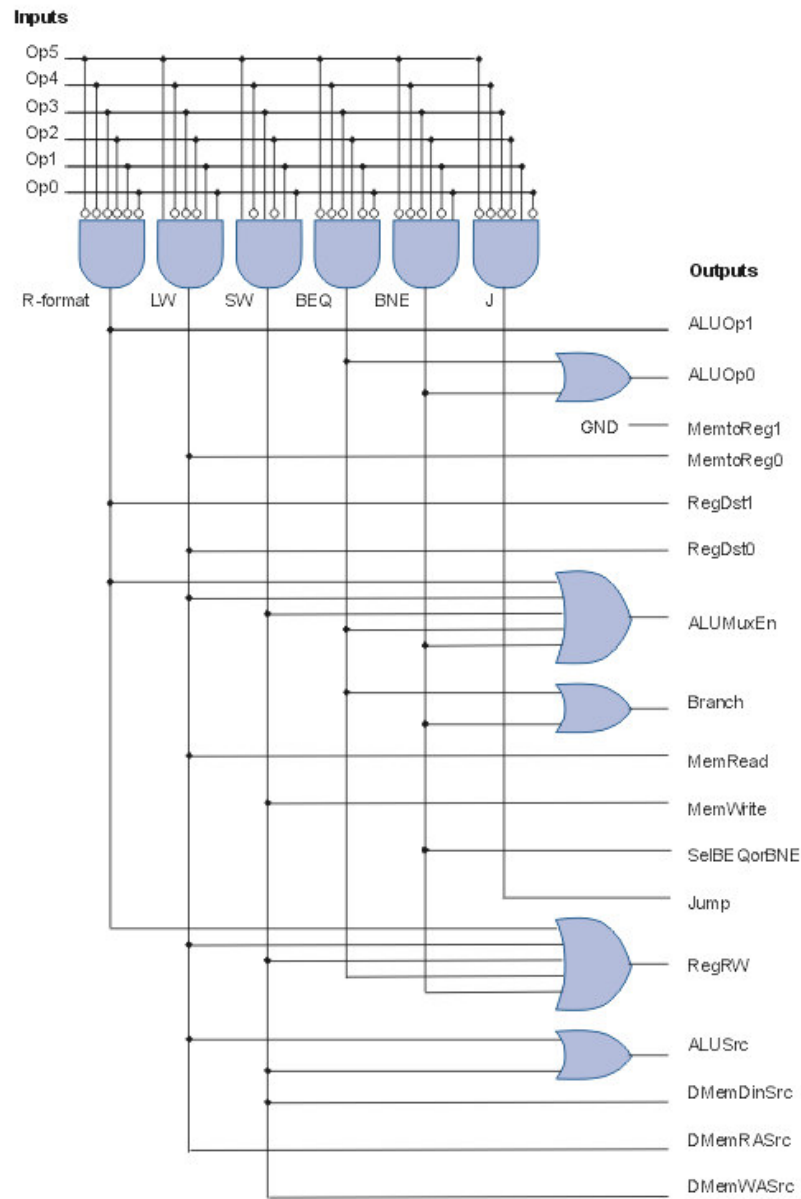


Figure C.9 Structured gate-level implementation of the Main Control Unit (adapted from [47]).

➤ Design Entry and Synthesis

Schematic Editor was used to create the design entry for the Main Control Unit shown in figure C.9. Figure C.10 shows the final schematic diagram.

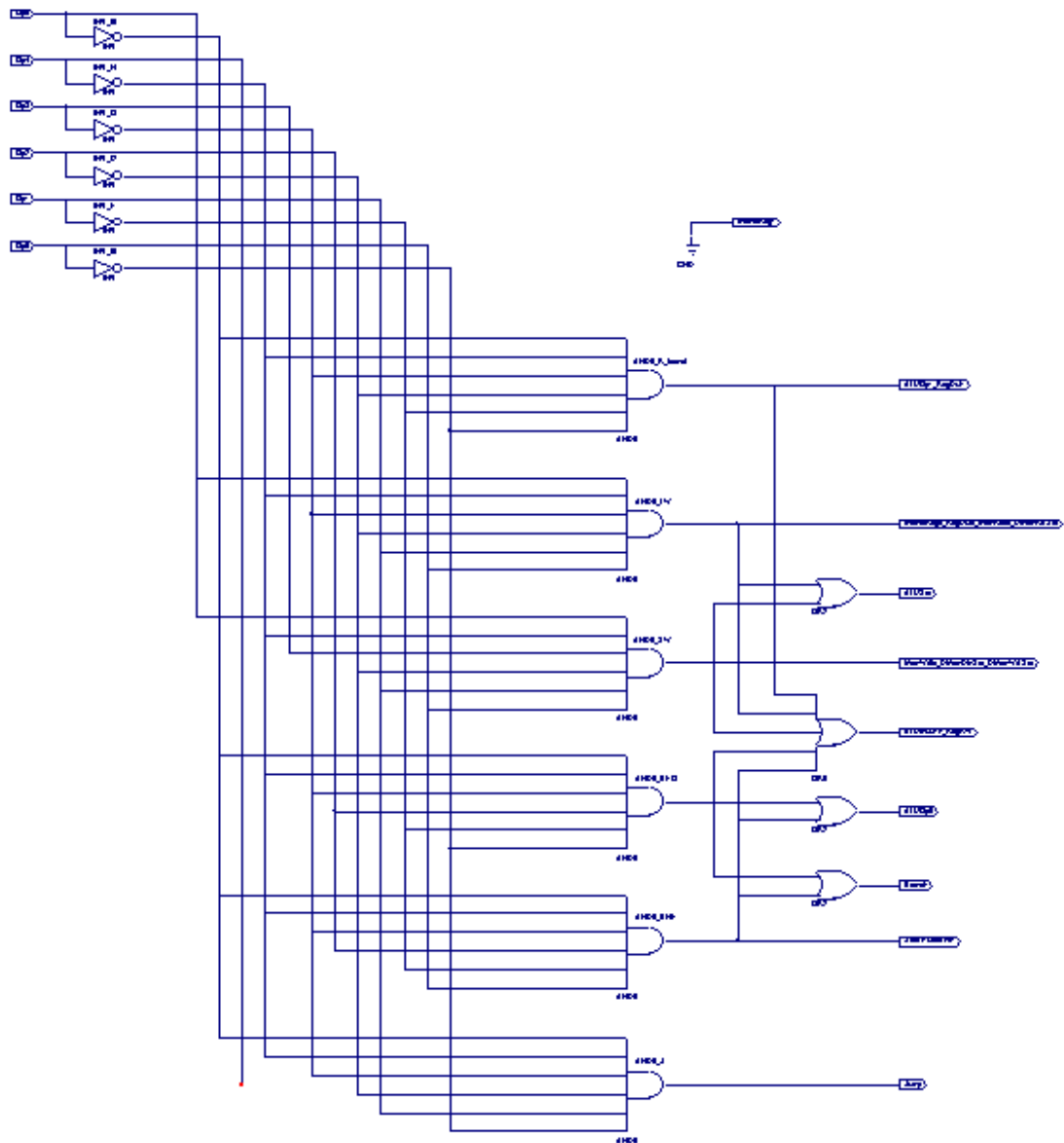


Figure C.10 Schematic diagram design entry in Schematic Editor for the Main Control Unit.

After synthesis of the schematic diagram in figure C.10 using XST, the following VHDL code was generated:

```
-- Vhdl model created from schematic C:\Xilinx\virtex2\data\drawing\and6.sch - Thu Oct
12 23:16:35 2006
```

```
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;
USE ieee.numeric_std.ALL;
-- synopsys translate_off
LIBRARY UNISIM;
USE UNISIM.Vcomponents.ALL;
-- synopsys translate_on
```

```
ENTITY AND6_MXILINX_main_control IS
PORT ( IO      : IN      STD_LOGIC;
```

```

        I1      :      IN      STD_LOGIC;
        I2      :      IN      STD_LOGIC;
        I3      :      IN      STD_LOGIC;
        I4      :      IN      STD_LOGIC;
        I5      :      IN      STD_LOGIC;
        O       :      OUT     STD_LOGIC);

end AND6_MXILINX_main_control;

ARCHITECTURE SCHEMATIC OF AND6_MXILINX_main_control IS
    SIGNAL I35      :      STD_LOGIC;
    SIGNAL O_DUMMY  :      STD_LOGIC;
    SIGNAL dummy    :      STD_LOGIC;

    ATTRIBUTE BOX_TYPE : STRING;
    ATTRIBUTE RLOC : STRING ;
    ATTRIBUTE RLOC OF I_36_93 : LABEL IS "X0Y0";
    ATTRIBUTE RLOC OF I_36_94 : LABEL IS "X0Y0";

    COMPONENT AND3
        PORT ( I0      :      IN      STD_LOGIC;
               I1      :      IN      STD_LOGIC;
               I2      :      IN      STD_LOGIC;
               O       :      OUT     STD_LOGIC);
    END COMPONENT;

    ATTRIBUTE BOX_TYPE OF AND3 : COMPONENT IS "BLACK_BOX";

    COMPONENT AND4
        PORT ( I0      :      IN      STD_LOGIC;
               I1      :      IN      STD_LOGIC;
               I2      :      IN      STD_LOGIC;
               I3      :      IN      STD_LOGIC;
               O       :      OUT     STD_LOGIC);
    END COMPONENT;

    ATTRIBUTE BOX_TYPE OF AND4 : COMPONENT IS "BLACK_BOX";

    COMPONENT FMAP
        PORT ( I1      :      IN      STD_LOGIC;
               I2      :      IN      STD_LOGIC;
               I3      :      IN      STD_LOGIC;
               I4      :      IN      STD_LOGIC;
               O       :      IN      STD_LOGIC);
    END COMPONENT;

    ATTRIBUTE BOX_TYPE OF FMAP : COMPONENT IS "BLACK_BOX";
BEGIN
    O <= O_DUMMY;

    I_36_69 : AND3
        PORT MAP (I0=>I3, I1=>I4, I2=>I5, O=>I35);

    I_36_85 : AND4
        PORT MAP (I0=>I0, I1=>I1, I2=>I2, I3=>I35, O=>O_DUMMY);

    I_36_93 : FMAP
        PORT MAP (I1=>I3, I2=>I4, I3=>I5, I4=>dummy, O=>I35);

    I_36_94 : FMAP
        PORT MAP (I1=>I0, I2=>I1, I3=>I2, I4=>I35, O=>O_DUMMY);

END SCHEMATIC;

-- Vhdl model created from schematic main_control.sch - Thu Oct 12 23:16:35 2006

LIBRARY ieee;
USE ieee.std_logic_1164.ALL;
USE ieee.numeric_std.ALL;
-- synopsys translate_off
LIBRARY UNISIM;
USE UNISIM.Vcomponents.ALL;
-- synopsys translate_on

```

```

ENTITY main_control IS
    PORT ( Op0      : IN      STD_LOGIC;
           Op1      : IN      STD_LOGIC;
           Op2      : IN      STD_LOGIC;
           Op3      : IN      STD_LOGIC;
           Op4      : IN      STD_LOGIC;
           Op5      : IN      STD_LOGIC;
           ALUMuxEn_RegRW : OUT   STD_LOGIC;
           ALUOp0    : OUT     STD_LOGIC;
           ALUOp1_RegDst1 : OUT   STD_LOGIC;
           ALUSrc    : OUT     STD_LOGIC;
           Branch    : OUT     STD_LOGIC;
           Jump      : OUT     STD_LOGIC;
           MemWrite_DMemDinSrc_DMemWASrc : OUT   STD_LOGIC;
           MemtoReg0_RegDst0_MemRead_DMemRASrc : OUT   STD_LOGIC;
           MemtoReg1  : OUT     STD_LOGIC;
           SelBEQorBNE : OUT     STD_LOGIC);

end main_control;

ARCHITECTURE SCHEMATIC OF main_control IS
    SIGNAL ALUOp1_RegDst1_DUMMY : STD_LOGIC;
    SIGNAL BEQ : STD_LOGIC;
    SIGNAL MemWrite_DMemDinSrc_DMemWASrc_DUMMY : STD_LOGIC;
    SIGNAL MemtoReg0_RegDst0_MemRead_DMemRASrc_DUMMY : STD_LOGIC;
    SIGNAL Op0_inv : STD_LOGIC;
    SIGNAL Op1_inv : STD_LOGIC;
    SIGNAL Op2_inv : STD_LOGIC;
    SIGNAL Op3_inv : STD_LOGIC;
    SIGNAL Op4_inv : STD_LOGIC;
    SIGNAL Op5_inv : STD_LOGIC;
    SIGNAL SelBEQorBNE_DUMMY : STD_LOGIC;

    ATTRIBUTE BOX_TYPE : STRING;
    ATTRIBUTE U_SET : STRING ;
    ATTRIBUTE U_SET OF AND6_R_format : LABEL IS "AND6_R_format_0";
    ATTRIBUTE U_SET OF AND6_LW : LABEL IS "AND6_LW_1";
    ATTRIBUTE U_SET OF AND6_SW : LABEL IS "AND6_SW_2";
    ATTRIBUTE U_SET OF AND6_BEQ : LABEL IS "AND6_BEQ_3";
    ATTRIBUTE U_SET OF AND6_BNE : LABEL IS "AND6_BNE_4";
    ATTRIBUTE U_SET OF AND6_J : LABEL IS "AND6_J_5";

    COMPONENT AND6_MXILINK_main_control
        PORT ( I0 : IN      STD_LOGIC;
              I1 : IN      STD_LOGIC;
              I2 : IN      STD_LOGIC;
              I3 : IN      STD_LOGIC;
              I4 : IN      STD_LOGIC;
              I5 : IN      STD_LOGIC;
              O : OUT     STD_LOGIC);
    END COMPONENT;

    COMPONENT GND
        PORT ( G : OUT     STD_LOGIC);
    END COMPONENT;

    ATTRIBUTE BOX_TYPE OF GND : COMPONENT IS "BLACK_BOX";
    COMPONENT INV
        PORT ( I : IN      STD_LOGIC;
              O : OUT     STD_LOGIC);
    END COMPONENT;

    ATTRIBUTE BOX_TYPE OF INV : COMPONENT IS "BLACK_BOX";
    COMPONENT OR2
        PORT ( I0 : IN      STD_LOGIC;
              I1 : IN      STD_LOGIC;
              O : OUT     STD_LOGIC);
    END COMPONENT;

    ATTRIBUTE BOX_TYPE OF OR2 : COMPONENT IS "BLACK_BOX";
    COMPONENT OR5
        PORT ( I0 : IN      STD_LOGIC;

```

```

        I1      :      IN      STD_LOGIC;
        I2      :      IN      STD_LOGIC;
        I3      :      IN      STD_LOGIC;
        I4      :      IN      STD_LOGIC;
        O       :      OUT      STD_LOGIC);
END COMPONENT;

ATTRIBUTE BOX_TYPE OF OR5 : COMPONENT IS "BLACK_BOX";
BEGIN
    ALUOp1_RegDst1 <= ALUOp1_RegDst1_DUMMY;
    MemWrite_DMemDinSrc_DMemWASrc <= MemWrite_DMemDinSrc_DMemWASrc_DUMMY;
    MemtoReg0_RegDst0_MemRead_DMemRASrc <=
        MemtoReg0_RegDst0_MemRead_DMemRASrc_DUMMY;
    SelBEQorBNE <= SelBEQorBNE_DUMMY;

    AND6_R_format : AND6_MXILINX_main_control
        PORT MAP (I0=>Op0_inv, I1=>Op1_inv, I2=>Op2_inv, I3=>Op3_inv,
            I4=>Op4_inv, I5=>Op5_inv, O=>ALUOp1_RegDst1_DUMMY);

    AND6_LW : AND6_MXILINX_main_control
        PORT MAP (I0=>Op0, I1=>Op1, I2=>Op2_inv, I3=>Op3_inv, I4=>Op4_inv,
            I5=>Op5, O=>MemtoReg0_RegDst0_MemRead_DMemRASrc_DUMMY);

    AND6_SW : AND6_MXILINX_main_control
        PORT MAP (I0=>Op0, I1=>Op1, I2=>Op2_inv, I3=>Op3, I4=>Op4_inv, I5=>Op5,
            O=>MemWrite_DMemDinSrc_DMemWASrc_DUMMY);

    AND6_BEQ : AND6_MXILINX_main_control
        PORT MAP (I0=>Op0_inv, I1=>Op1_inv, I2=>Op2, I3=>Op3_inv, I4=>Op4_inv,
            I5=>Op5_inv, O=>BEQ);

    AND6_BNE : AND6_MXILINX_main_control
        PORT MAP (I0=>Op0, I1=>Op1_inv, I2=>Op2, I3=>Op3_inv, I4=>Op4_inv,
            I5=>Op5_inv, O=>SelBEQorBNE_DUMMY);

    AND6_J : AND6_MXILINX_main_control
        PORT MAP (I0=>Op0_inv, I1=>Op1, I2=>Op2_inv, I3=>Op3_inv, I4=>Op4_inv,
            I5=>Op5_inv, O=>Jump);

    XLXI_23 : GND
        PORT MAP (G=>MemtoReg1);

    INV_i4 : INV
        PORT MAP (I=>Op4, O=>Op4_inv);

    INV_i3 : INV
        PORT MAP (I=>Op3, O=>Op3_inv);

    INV_i2 : INV
        PORT MAP (I=>Op2, O=>Op2_inv);

    INV_i1 : INV
        PORT MAP (I=>Op1, O=>Op1_inv);

    INV_i0 : INV
        PORT MAP (I=>Op0, O=>Op0_inv);

    INV_i5 : INV
        PORT MAP (I=>Op5, O=>Op5_inv);

    OR2_i1 : OR2
        PORT MAP (I0=>MemWrite_DMemDinSrc_DMemWASrc_DUMMY,
            I1=>MemtoReg0_RegDst0_MemRead_DMemRASrc_DUMMY, O=>ALUSrc);

    XLXI_24 : OR2
        PORT MAP (I0=>SelBEQorBNE_DUMMY, I1=>BEQ, O=>ALUOp0);

    XLXI_27 : OR2
        PORT MAP (I0=>SelBEQorBNE_DUMMY, I1=>BEQ, O=>Branch);

    OR5_i1 : OR5
        PORT MAP (I0=>SelBEQorBNE_DUMMY, I1=>BEQ,
            I2=>MemWrite_DMemDinSrc_DMemWASrc_DUMMY,

```

```

I3=>MementoReg0_RegDst0_MemRead_DMemRASrc_DUMMY, I4=>ALUOp1_RegDst1_DUMMY,
O=>ALUMuxEn_RegRW);

END SCHEMATIC;

```

➤ Synthesis Results

Using the Xilinx ISE synthesis tools, the hardware implementation for the Main Control Unit, was generated. Figure C.11 shows the resulting top level RTL symbol while figure C.12 shows the resulting top level RTL schematic diagram.

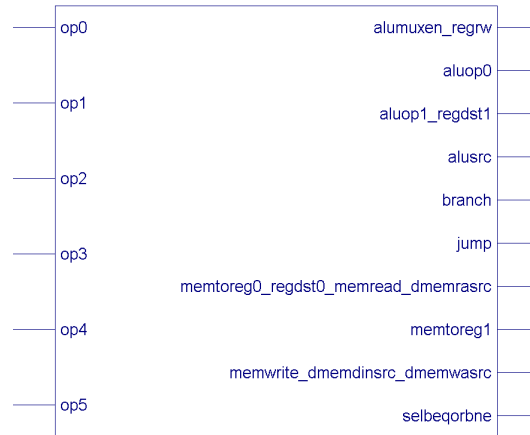


Figure C.11 Resulting top level RTL symbol for the synthesized Main Control Unit.

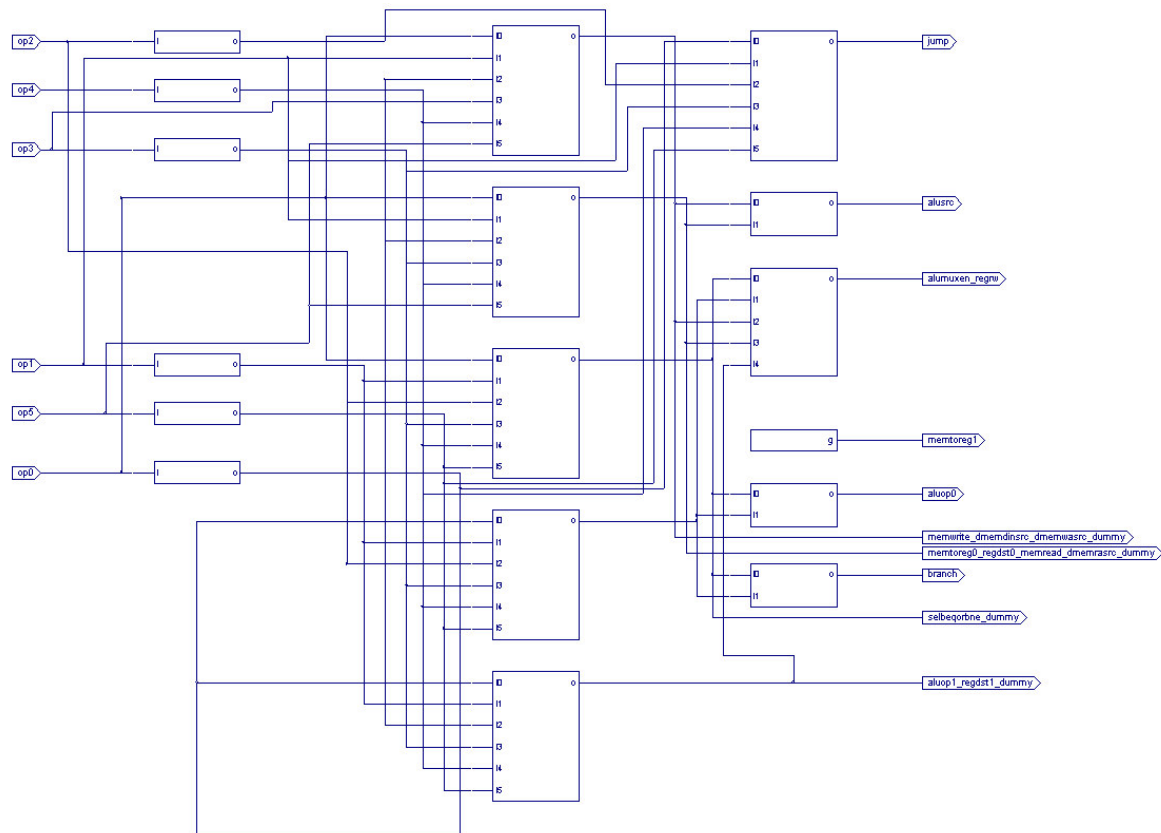


Figure C.12 Resulting top level RTL schematic for the synthesized Main Control Unit.

➤ FPGA Device Synthesis Summary

After the hardware implementation for the Main Control Unit using the Xilinx ISE synthesis tools, the Synthesis Report was generated. The most important FPGA Device Synthesis Statistics from this report, are shown below:

```
Design Statistics:
# I/Os                                : 16

Cell Usage:
# BELS                                : 29
# and3                                : 6
# and4                                : 6
# GND                                  : 7
# inv                                  : 6
# or2                                  : 3
# or5                                  : 1
# IO Buffers                           : 16
# IBUF                                 : 6
# OBUF                                 : 10
# Others                               : 12
# fmap                                : 12
```

Device utilization summary:

```
Number of bonded IOBs:                16 out of 1108    1%
```

➤ *Place-and-Route onto the FPGA*

In figure C.13, FPGA Editor shows the synthesized Main Control Unit after place-and-route onto the target Virtex-II FPGA chip. Notice that these are the blue interconnections concentrated at the lower left corner of the FPGA chip.

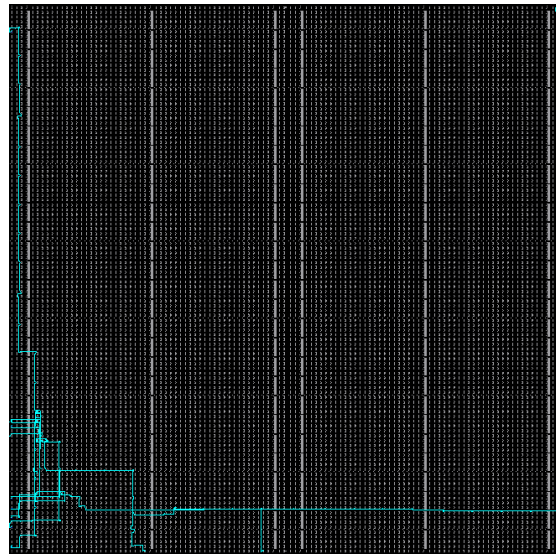


Figure C.13 FPGA Editor showing the synthesized Main Control Unit.

➤ *Simulation Results*

Figure C.14 shows the waveform results of simulating the Main Control Unit VHDL behavioural model in Mentor Graphics ModelSim. These waveforms are in binary format. It is clear that the resulting synthesized hardware functions according to the specified behaviour of the Main Control Unit outlined in the truth table of figure C.8. This concludes the design cycle for this component.

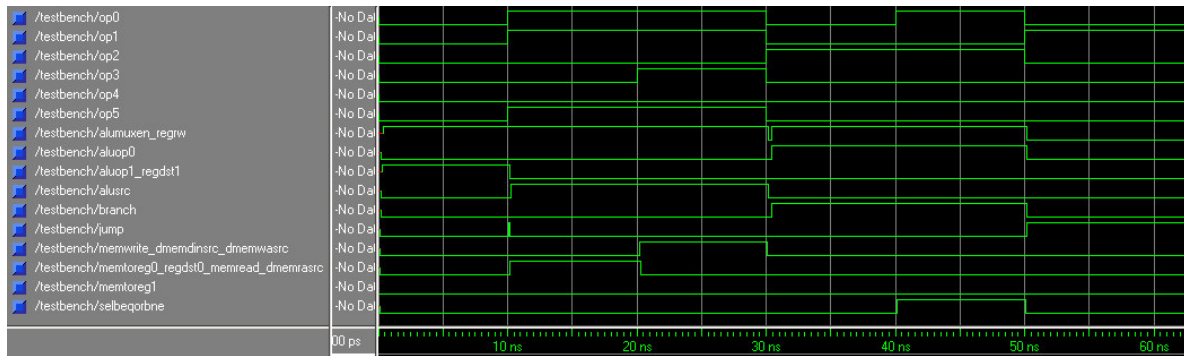


Figure C.14 Results of simulating the synthesized Main Control Unit, using ModelSim.

C.3 Summary and Conclusions

This appendix covered the design, synthesis, and simulation of the control unit tailored to the requirements of this research.