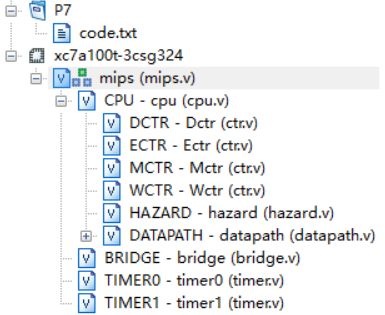
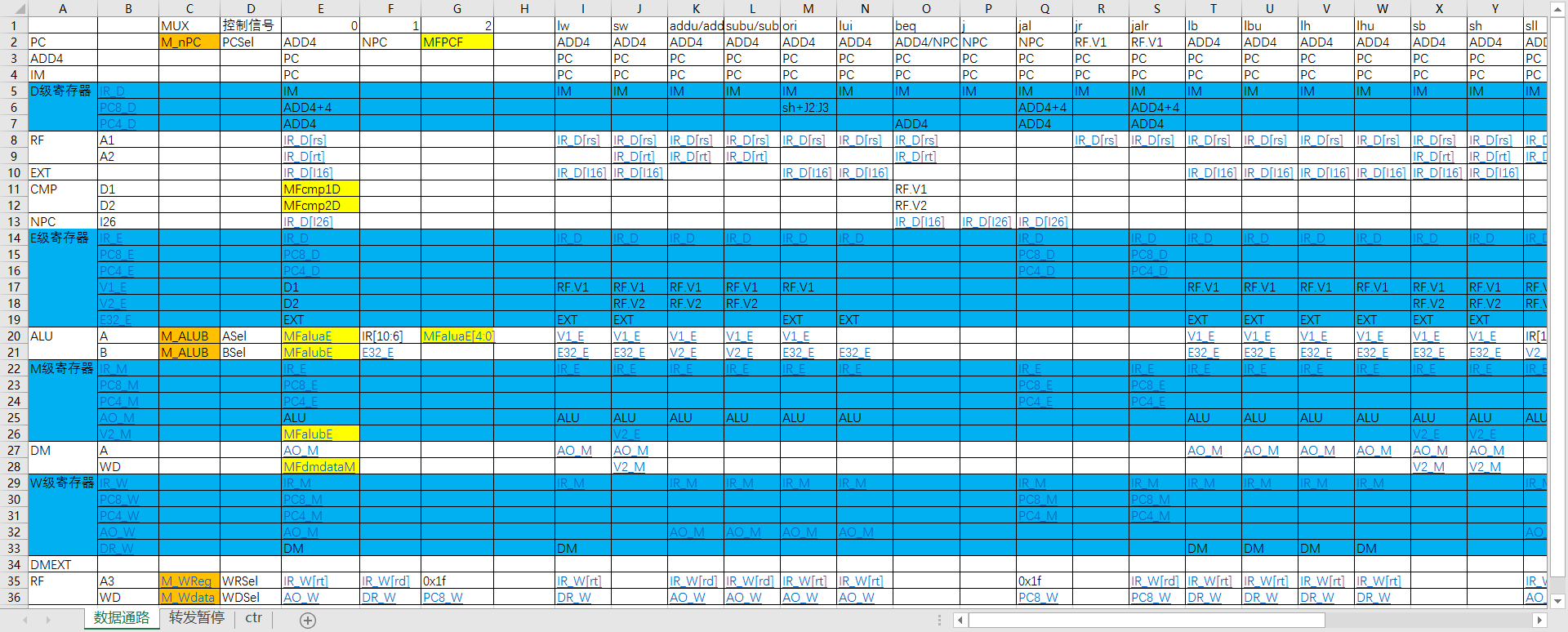
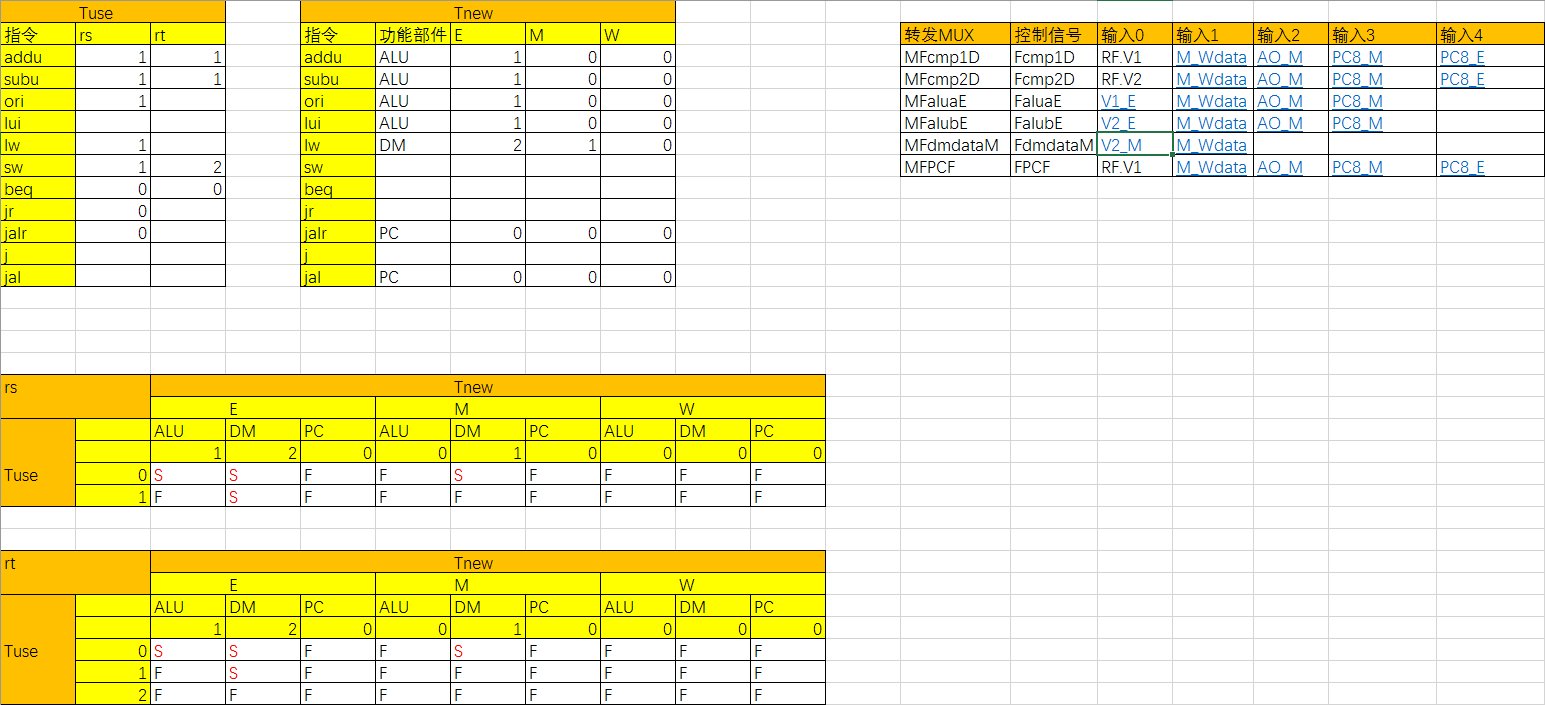
计算机组成原理实验报告

1. CPU设计文档
2. 总体设计



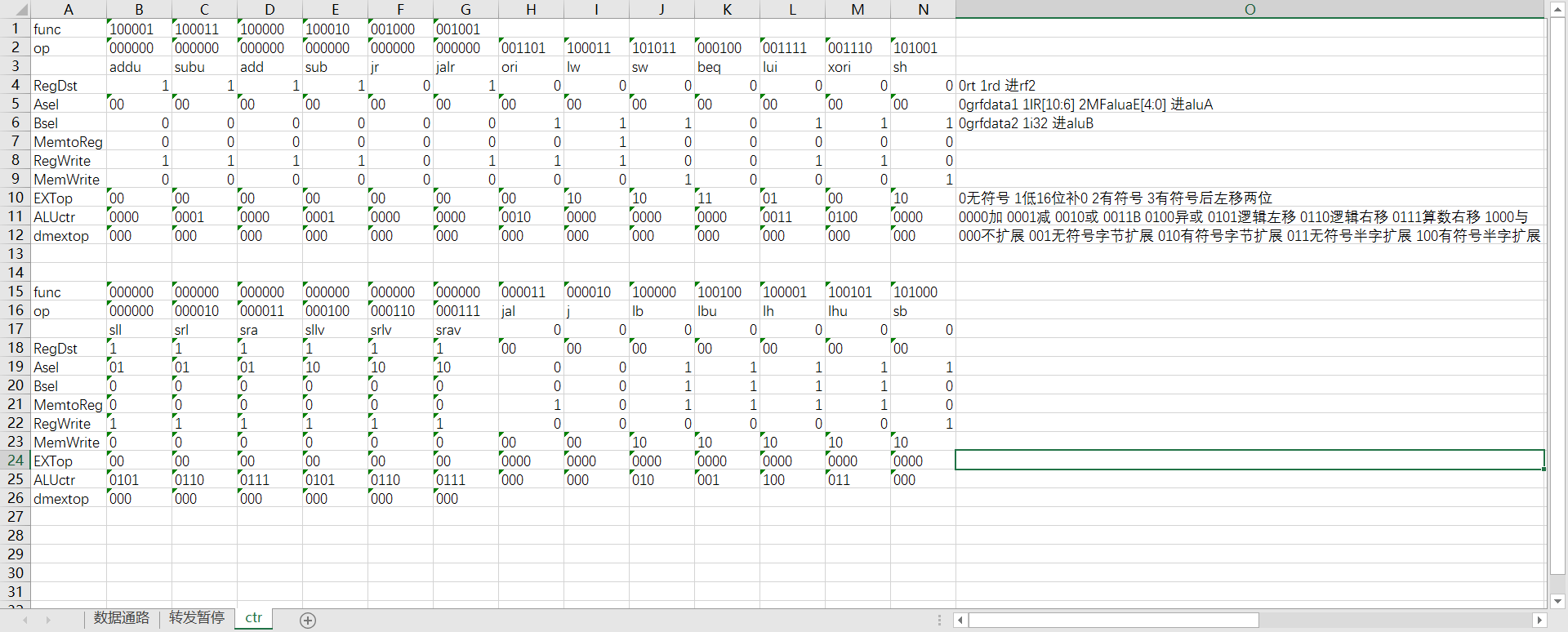
图表 1模块设计



图表 2数据通路设计

图表 3 转发暂停设计

图表 4控制器设计

1. 数据通路设计
2. datapath（数据通路）
3. 端口说明

表格 1datapath端口说明

|  |  |  |
| --- | --- | --- |
| 信号名 | 方向 | 描述 |
| clk | I | 时钟信号 |
| reset | I | 复位信号1：复位0：无效 |
| PCsel[1:0] | I | D控制器发来选择nPC信号10：转发器MFPCF结果01：NPC 00：ADD4 |
| i16[15:0] | I | D控制器发来16位立即数 |
| i26[25:0] | I | D控制器发来26位立即数 |
| RegWrite | I | W控制器发来写寄存器信号1：写寄存器0：无效 |
| MemWrite | I | M控制器发来写内存信号1：写内存0：无效 |
| EXTop[2:0] | I | D控制器发来扩展器信号000：无符号扩展 001：低16位补0 010：有符号扩展 011：有符号扩展后逻辑左移两位 |
| ALUctr[2:0] | I | E控制器发来ALU控制信号000：加运算001：减运算010：或运算011：输出写入数据2 100:异或运算 |
| WRsel[1:0] | I | D控制器发来选择寄存器A3端口信号10:31号寄存器01：IR\_W[rd] 00：IR\_W[rt] |
| WDsel[1:0] | I | W控制器发来选择寄存器输入信号00：AO\_W01：DR\_W10：PC8\_W |
| Fcmp1D[2:0] | I | 冒险单元发来选择MFcmp1D信号000：RF.V1 001：M\_Wdata 010：AO\_M 011：PC8\_M 100: PC8\_E |
| Fcmp2D[2:0] | I | 冒险单元发来选择MFcmp2D信号 000：RF.V2 001：M\_Wdata 010：AO\_M 011：PC8\_M 100: PC8\_E |
| FdmdataM | I | 冒险单元发来选择MFdmdataM信号 0：V2\_M 1：M\_Wdata |
| FaluaE[1:0] | I | 冒险单元发来选择MFaluaE信号 00：V1\_E 01：M\_Wdata 10：AO\_M 11：PC8\_M |
| FalubE[1:0] | I | 冒险单元发来选择MFalubE信号 00：V2\_E 01：M\_Wdata 10：AO\_M 11：PC8\_M |
| FPCF[2:0] | I | 冒险单元发来选择MFPCF信号 000：RF.V1 001：M\_Wdata 010：AO\_M 011：PC8\_M 100: PC8\_E |
| Bsel | I | E控制器发来ALUb选择信号 0：MFalubE选择结果 1：E32\_E |
| stall | I | 冒险单元发来选择暂停信号 1：暂停 0：无效 |
| beq | I | D控制器发来beq识别信号 1：beq 0：无效 |
| IRF[31:0] | O | 输出到D控制器的F级指令 |
| IRD[31:0] | O | 输出到E控制器的D级指令 |
| IRE[31:0] | O | 输出到M控制器的E级指令 |
| IRM[31:0] | O | 输出到W控制器的M级指令 |
| RESE[1:0] | O | 输出到冒险单元的E级Tnew状态 00：NW不写 01：写ALU 10：写DM 11：写PC |
| RESM[1:0] | O | 输出到冒险单元的M级Tnew状态 00：NW不写 01：写ALU 10：写DM 11：写PC |
| RESW[1:0] | O | 输出到冒险单元的W级Tnew状态 00：NW不写 01：写ALU 10：写DM 11：写PC |
| RFA3E[4:0] | O | 输出到冒险单元的E级指令A3寄存器 |
| RFA3M[4:0] | O | 输出到冒险单元的M级指令A3寄存器 |
| RFA3W[4:0] | O | 输出到冒险单元的W级指令A3寄存器 |

1. 功能定义

表格 2datapath功能定义

|  |  |  |
| --- | --- | --- |
| 序号 | 功能名称 | 功能描述 |
| 1 | 连接基本模块 | 通过datapath，以声明中间变量和实例化引用的方式连接各基础模块 |

1. ifu（取指令单元）
2. 端口说明

表格 3ifu端口说明

|  |  |  |
| --- | --- | --- |
| 信号名 | 方向 | 描述 |
| reset | I | 复位信号 1：复位 0：无效 |
| clk | I | 时钟信号 |
| PCsel[1:0] | I | D控制器发来选择nPC信号 10：转发器MFPCF结果 01：NPC 00：ADD4 |
| CO | I | cmp发来比较信号 1：alu两输入相等 0：alu两输入不等 |
| i16 [15:0] | I | D控制器发来16位立即数 |
| i26[25:0] | I | D控制器发来26位立即数 |
| PCtempD[31:0] | I | MFPCF转发多选器的结果 |
| beq | I | D控制器发来beq识别信号 1：beq 0：无效 |
| stall | I | 冒险单元发来选择暂停信号 1：暂停 0：无效 |
| instruction[31:0] | O | 输出的指令 |
| PC8[31:0] | O | 输出的当前PC+8 |

1. 功能定义

表格 4 ifu功能定义

|  |  |  |
| --- | --- | --- |
| 序号 | 功能名称 | 功能描述 |
| 1 | 复位 | 复位信号有效时，PC被置为0x00000000 |
| 2 | 取指令 | 根据PC从IM中取出指令 |
| 3 | 计算下条指令地址 | PCPC+4 || PCreg1data || PCPC + 4 + immed32 || PC{PC[31:28], immed26, 2'b0} |
| 4 | 暂停 | Stall信号有效时，冻结PC寄存器 |

1. rf（寄存器堆）
2. 端口说明

表格 5grf端口说明

|  |  |  |
| --- | --- | --- |
| 信号名 | 方向 | 描述 |
| reset | I | 复位信号 1：复位 0：无效 |
| clk | I | 时钟信号 |
| reg1 [4:0] | I | 读寄存器号1编号 |
| Reg2 [4:0] | I | 读寄存器号2编号 |
| writereg[4:0] | I | 写寄存器编号 |
| regwrite | I | 写控制信号 1：写入 0：无效 |
| writedata[31:0] | I | 写入的32位数据 |
| data1[31:0] | O | 32位寄存器1输出 |
| data2[31:0] | O | 32位寄存器2输出 |

1. 功能定义

表格 6grf功能定义

|  |  |  |
| --- | --- | --- |
| 序号 | 功能名称 | 功能描述 |
| 1 | 复位 | 复位信号有效时，32个寄存器被置为0x00000000 |
| 2 | 写寄存器 | 写寄存器控制信号有效时，把32位数据写入寄存器 |
| 3 | 读寄存器 | 根据输入的地址读出两个寄存器中的值 |

1. alu（算术逻辑单元）
2. 端口说明

表格 7alu端口说明

|  |  |  |
| --- | --- | --- |
| 信号名 | 方向 | 描述 |
| A[31:0] | I | 32位写入数据1 |
| B[31:0] | I | 32位写入数据2 |
| ALUOp[2:0] | I | 控制信号 000：加运算 001：减运算 010：或运算 011：输出写入数据2 100:异或运算 |
| AO[31:0] | O | 32位输出数据 |

1. 功能定义

表格 8alu功能定义

|  |  |  |
| --- | --- | --- |
| 序号 | 功能名称 | 功能描述 |
| 1 | 加运算 | A+B |
| 2 | 减运算 | A-B |
| 3 | 或运算 | A|B |
| 4 | 输出写入数据2 | B |
| 5 | 异或运算 | A^B |

1. dm（数据存储器）
2. 端口说明

表格 9dm端口说明

|  |  |  |
| --- | --- | --- |
| 信号名 | 方向 | 描述 |
| D1[31:0] | I | 32位输入数据1 |
| D2[31:0] | I | 32位输入数据2 |
| CO | O | 比较结果 0：不相等 1：相等 |

1. 功能定义

表格 10dm功能定义

|  |  |  |
| --- | --- | --- |
| 序号 | 功能名称 | 功能描述 |
| 1 | 比较 | 比较两输入数据大小，相等输出1，否则输出0 |

1. cmp（比较器）
2. 端口说明

表格 11dm端口说明

|  |  |  |
| --- | --- | --- |
| 信号名 | 方向 | 描述 |
| clk | I | 时钟信号 |
| reset | I | 复位信号 1：复位 0：无效 |
| ADDR [31:0] | I | 32位写入内存地址 |
| din[31:0] | I | 32位写入数据 |
| PC8[31:0] | I | 当前PC+8 |
| MemWrite | I | 写内存控制信号 1：写入 0：无效 |
| dout[31:0] | O | 32位输出数据 |

1. 功能定义

表格 12dm功能定义

|  |  |  |
| --- | --- | --- |
| 序号 | 功能名称 | 功能描述 |
| 1 | 复位 | 复位信号有效时，内存和读出内存的寄存器被置为0x00000000 |
| 2 | 写内存 | 写内存控制信号有效时，根据输入的地址写入32位数据 |
| 3 | 读内存 | 根据输入的地址读出内存数据 |

1. ext（位扩展器）
2. 端口说明

表格 13ext端口说明

|  |  |  |
| --- | --- | --- |
| 信号名 | 方向 | 描述 |
| immed16[15:0] | I | 16位写入立即数 |
| EXTop[2:0] | I | 扩展控制信号 000：无符号扩展 001：低16位补0 010：有符号扩展 011：有符号扩展后逻辑左移两位 |
| EO[31:0] | O | 32位输出立即数 |

1. 功能定义

表格 14ext功能定义

|  |  |  |
| --- | --- | --- |
| 序号 | 功能名称 | 功能描述 |
| 1 | 无符号扩展 | 高16位补0 |
| 2 | 低16位补0 | 低16位补0 |
| 3 | 有符号扩展 | Immed[15]为1时高16位补1，为0时高16位补0 |
| 4 | 有符号扩展后逻辑左移两位 | Immed[15]为1时高16位补1，为0时高16位补0，再左移两位，溢出舍去，低2位补0 |

1. mux（多路选择器）
2. 端口说明

表格 15mux端口说明

|  |  |  |
| --- | --- | --- |
| 信号名 | 方向 | 描述 |
| A[4:0] | I | 5位输入A |
| B[4:0] | I | 5位输入B |
| C[4:0] | I | 5位输入C |
| Op[1:0] | I | 选择控制信号 10：输出C 01：输出B 00：输出A |
| O [4:0] | O | 5位输出O |
| A[31:0] | I | 32位输入A |
| B[31:0] | I | 32位输入B |
| C[31:0] | I | 32位输入C |
| D[31:0] | I | 32位输入D |
| E[31:0] | I | 32位输入E |
| op[2:0] | I | 选择控制信号 100：输出E 011：输出D 010：输出C 001：输出B 000：输出A |
| O[31:0] | O | 32位输出O |

1. 功能定义

表格 16mux功能定义

|  |  |  |
| --- | --- | --- |
| 序号 | 功能名称 | 功能描述 |
| 1 | 5位输入3选1 | option为10输出C，为01输出B，为00输出A |
| 2 | 32位输入2选1 | option为1输出B，为0输出A |
| 3 | 32位输入3选1 | option为10输出C，为01输出B，为00输出A |
| 4 | 32位输入4选1 | option为11输出D，为10输出C，为01输出B，为00输出A |
| 5 | 32位输入5选1 | option100输出E，011输出D，010输出C，001输出B，000输出A |

1. Dregs（D级流水线寄存器）
2. 端口说明

表格 17 Dregs端口说明

|  |  |  |
| --- | --- | --- |
| 信号名 | 方向 | 描述 |
| clk | I | 时钟信号 |
| reset | I | 复位信号 1：复位0：无效 |
| IR[31:0] | I | D级部件将使用的指令 |
| PC8[31:0] | I | D级部件对应指令的PC+8 |
| Stall | I | 冒险单元输入的暂停信号 |
| PCsel[1:0] | I | 将传回F级部件的PC选择信号 |
| I16[15:0] | I | 将传回F级部件的16位立即数 |
| I26[25:0] | I | 将传回F级部件的26位立即数 |
| Beq | I | 将传回F级部件的beq识别信号 |
| EXTop[2:0] | I | 扩展控制信号 000：无符号扩展 001：低16位补0 010：有符号扩展 011：有符号扩展后逻辑左移两位 |
| WRsel[1:0] | I | 选择寄存器A3端口信号 10:31号寄存器 01：IR\_W[rd] 00：IR\_W[rt] |
| IR\_D [31:0] | O | 输出到E级寄存器的指令 |
| PC8\_D[31:0] | O | 输出到E级寄存器的PC+8 |
| PCsel\_D[1:0] | O | 输出到F级寄存器的PC选择信号 |
| i16\_D | O | 将传回F级部件的16位立即数 |
| I26\_D | O | 将传回F级部件的26位立即数 |
| Beq\_D | O | 将传回F级部件的beq识别信号 |
| EXTop\_D[2:0] | O | 扩展控制信号 000：无符号扩展 001：低16位补0 010：有符号扩展 011：有符号扩展后逻辑左移两位 |
| WRsel\_D[1:0] | O | 选择寄存器A3端口信号 10:31号寄存器 01：IR\_W[rd] 00：IR\_W[rt] |

1. 功能定义

表格 18 Dregs功能定义

|  |  |  |
| --- | --- | --- |
| 序号 | 功能名称 | 功能描述 |
| 1 | 存储结果 | 存储F级部件结果，发送到D级部件或F级部件 |

1. Eregs（E级流水线寄存器）
2. 端口说明

表格 19 Eregs端口说明

|  |  |  |
| --- | --- | --- |
| 信号名 | 方向 | 描述 |
| clk | I | 时钟信号 |
| reset | I | 复位信号 1：复位 0：无效 |
| IR[31:0] | I | E级部件将使用的指令 |
| PC8[31:0] | I | E级部件对应指令的PC+8 |
| Stall | I | 冒险单元输入的暂停信号 |
| V1[31:0] | I | MFcmp1D转发而来的结果 |
| V2[31:0] | I | MFcmp2D转发而来的结果 |
| E32[31:0] | I | EXT结果 |
| RFA3[4:0] | I | E级指令要写入的寄存器编号 |
| Bsel | I | Alub多选器的选择信号 |
| ALUctr [2:0] | I | 控制信号 000：加运算 001：减运算 010：或运算 011：输出写入数据2 100:异或运算 |
| IR\_E[31:0] | O | 输出到E级寄存器的指令 |
| PC8\_E[31:0] | O | 输出到E级寄存器的PC+8 |
| RFA3E[4:0] | O | 输出到E级的要写入的寄存器编号 |
| V1[31:0] | O | MFcmp1D转发而来的结果 |
| V2[31:0] | O | MFcmp2D转发而来的结果 |
| Bsel\_E | O | Alub多选器的选择信号 |
| ALUctr\_E[2:0] | O | alu控制信号 000：加运算 001：减运算 010：或运算 011：输出写入数据2 100:异或运算 |
| Res\_E [1:0] | O | E级指令对部件的产生结果位置 ALU：在alu产生结果 DM：在dm产生结果 PC：产生PC结果 NW：nowrite，不产生结果 |

1. 功能定义

表格 20Eregs功能定义

|  |  |  |
| --- | --- | --- |
| 序号 | 功能名称 | 功能描述 |
| 1 | 存储结果 | 存储D级部件结果，发送到E级部件 |
| 2 | 产生控制转发信号 | 计算E级指令对部件的产生结果位置 |

1. Mregs（M级流水线寄存器）
2. 端口说明

表格 21Mregs端口说明

|  |  |  |
| --- | --- | --- |
| 信号名 | 方向 | 描述 |
| clk | I | 时钟信号 |
| reset | I | 复位信号 1：复位 0：无效 |
| IR[31:0] | I | M级部件将使用的指令 |
| PC8[31:0] | I | M级部件对应指令的PC+8 |
| AO[31:0] | I | ALU的结果 |
| V2[31:0] | I | MFcmp2D转发而来的结果 |
| RFA3[4:0] | I | M级指令要写入的寄存器编号 |
| MemWrite | I | Dm写入控制信号 1：写入 0：无效 |
| IR\_M[31:0] | O | 输出到W级寄存器的指令 |
| PC8\_M[31:0] | O | 输出到W级寄存器的PC+8 |
| RFA3M[4:0] | O | 输出到M级的要写入的寄存器编号 |
| AO\_M[31:0] | O | 输出到W级的alu结果 |
| V2[31:0] | O | 输出到W级的MFcmp2D转发而来的结果 |
| MemWrite | O | 输出到M级部件的Dm写入控制信号 1：写入 0：无效 |
| Res\_M[1:0] | O | M级指令对部件的产生结果位置 ALU：在alu产生结果 DM：在dm产生结果 PC：产生PC结果 NW：nowrite，不产生结果 |

1. 功能定义

表格 22Mregs功能定义

|  |  |  |
| --- | --- | --- |
| 序号 | 功能名称 | 功能描述 |
| 1 | 存储结果 | 存储E级部件结果，发送到M级部件 |
| 2 | 产生控制转发信号 | 计算M级指令对部件的产生结果位置 |

1. Wregs（W级流水线寄存器）
2. 端口说明

表格 23Wregs端口说明

|  |  |  |
| --- | --- | --- |
| 信号名 | 方向 | 描述 |
| clk | I | 时钟信号 |
| reset | I | 复位信号 1：复位 0：无效 |
| IR[31:0] | I | W级部件将使用的指令 |
| PC8[31:0] | I | W级部件对应指令的PC+8 |
| AO[31:0] | I | ALU的结果 |
| DR[31:0] | I | DM的结果 |
| RFA3[4:0] | I | W级指令要写入的寄存器编号 |
| RegWrite | I | 寄存器堆写入控制信号 1：写入 0：无效 |
| WDsel[1:0] | I | 寄存器堆写入数据选择信号 10: PC8\_W 01：DR\_W 00：AO\_W |
| PC8\_W[31:0] | O | 输出到W级部件的PC+8 |
| RFA3W[4:0] | O | 输出到W级的要写入的寄存器编号 |
| AO\_W[31:0] | O | 输出到W级部件的alu结果 |
| DR\_W[31:0] | O | 输出到W级部件DM的结果 |
| RegWrite | O | 输出到W级部件的寄存器写入控制信号 1：写入 0：无效 |
| WDsel\_W[1:0] | O | 寄存器堆写入数据选择信号 10: PC8\_W 01：DR\_W 00：AO\_W |
| Res\_W[1:0] | O | W级指令对部件的产生结果位置 ALU：在alu产生结果 DM：在dm产生结果 PC：产生PC结果 NW：nowrite，不产生结果 |

1. 功能定义

表格 24Wregs功能定义

|  |  |  |
| --- | --- | --- |
| 序号 | 功能名称 | 功能描述 |
| 1 | 存储结果 | 存储M级部件结果，发送到W级部件 |
| 2 | 产生控制转发信号 | 计算W级指令对部件的产生结果位置 |

1. Cp0（协处理器）
2. 端口说明

表格 25cp0端口说明

|  |  |  |
| --- | --- | --- |
| 信号名 | 方向 | 描述 |
| clk | I | 时钟信号 |
| reset | I | 复位信号 1：复位 0：无效 |
| A1[4:0] | I | 读cp0寄存器号 |
| A2[4:0] | I | 写cp0寄存器号 |
| DIn[31:0] | I | 写入数据 |
| PC[31:0] | I | 异常PC |
| BD | I | 延迟槽指示信号 1：在延迟槽中 0：无效 |
| ExcCode [6:2] | I | 中断异常类型 0：中断 4：取指取数错误 5：存数地址错误 10：无效指令 12：自陷式指令溢出 |
| HWInt[5:0] | I | 6个中断设备 |
| We | I | 写使能信号 1：写入 0：无效 |
| EXLClr | I | EXL清零信号 1：清零 0：无效 |
| AdEL | I | AdEL指示信号 |
| AdES | I | AdES指示信号 |
| IntReq | O | 中断指示 1：中断 0：无效 |
| EPC[31:2] | O | 异常PC |
| DOut[31:0] | O | Cp0输出数据 |

1. 功能定义

表格 26cp0功能定义

|  |  |  |
| --- | --- | --- |
| 序号 | 功能名称 | 功能描述 |
| 1 | 复位 | 复位信号有效时，HI和LO寄存器和信号被置为0 |
| 2 | 写入数据 | 向cp0中寄存器写入数据din |
| 3 | 读取数据 | 读出cp0中寄存器值 |
| 4 | 输出中断信号 | 输出中断信号 |
| 5 | 存储中断异常类型 | 在寄存器中存储中断异常类型 |

1. Muldiv（乘除单元）
2. 端口说明

表格 27Muldiv端口说明

|  |  |  |
| --- | --- | --- |
| 信号名 | 方向 | 描述 |
| clk | I | 时钟信号 |
| reset | I | 复位信号 1：复位 0：无效 |
| A[31:0] | I | 乘除单元输入数据1 |
| B[31:0] | I | 乘除单元输入数据2 |
| op[2:0] | I | 乘除单元控制信号 000：无效 001：有符号乘 010：无符号乘 011：有符号除 100：无符号除 101：mthi 110：mtlo |
| HI[31:0] | O | HI寄存器 |
| LO[31:0] | O | LO寄存器 |
| busy | O | 乘除运算进行信号 1：正在运算 0：无效 |
| start | O | Busy信号产生信号 1：下一个时钟上升沿busy有效 0：无效 |

1. 功能定义

表格 28Muldiv功能定义

|  |  |  |
| --- | --- | --- |
| 序号 | 功能名称 | 功能描述 |
| 1 | 复位 | 复位信号有效时，HI和LO寄存器和信号被置为0 |
| 2 | 有符号乘 | 有符号乘法，其中结果高32位保存在HI，低32位保存在LO，时长5周期 |
| 3 | 无符号乘 | 无符号乘法，其中结果高32位保存在HI，低32位保存在LO，时长5周期 |
| 4 | 有符号除 | 有符号除法，其中结果余数保存在HI，商保存在LO，时长10周期 |
| 5 | 无符号除 | 无符号除法，其中结果余数保存在HI，商保存在LO，时长10周期 |
| 6 | 修改HI | 将rs的值存入HI寄存器 |
| 7 | 修改LO | 将rs的值存入LO寄存器 |
| 8 | 读取HI | 读取HI寄存器的值，写入rd |
| 9 | 读取LO | 读取LO寄存器的值，写入rd |

1. 冒险单元
2. 端口说明

表格 29hazard端口说明

|  |  |  |
| --- | --- | --- |
| 信号名 | 方向 | 描述 |
| IR[31:0] | I | D级部件将使用的指令 |
| Res\_E[1:0] | I | E级传来的控制信号 |
| Res\_M[1:0] | I | M级传来的控制信号 |
| Res\_W[1:0] | I | W级传来的控制信号 |
| A3\_E[4:0] | I | E级指令要写入的寄存器编号 |
| A3\_M[4:0] | I | M级指令要写入的寄存器编号 |
| A3\_W[4:0] | I | W级指令要写入的寄存器编号 |
| A1\_D[4:0] | I | D级指令要读入的寄存器编号1 |
| A2\_D[4:0] | I | D级指令要读入的寄存器编号2 |
| A1\_E[4:0] | I | E级指令要读入的寄存器编号1 |
| A2\_E[4:0] | I | E级指令要读入的寄存器编号2 |
| A2\_M[4:0] | I | M级指令要读入的寄存器编号 |
| Busy | I | 乘除运算进行信号 1：正在运算 0：无效 |
| start | I | Busy信号产生信号 1：下一个时钟上升沿busy有效 0：无效 |
| stall | O | 暂停信号 1：暂停 0：无效 |
| Fcmp1D[2:0] | O | 输出到D级的cmp编号1转发信号 000：RF.V1 001：M\_Wdata 010：AO\_M 011：PC8\_M 100: PC8\_E |
| Fcmp2D[2:0] | O | 输出到D级的cmp编号2转发信号 000：RF.V2 001：M\_Wdata 010：AO\_M 011：PC8\_M 100: PC8\_E |
| FaluaE[1:0] | O | 输出到E级ALUa的转发信号 00：V1\_E 01：M\_Wdata 10：AO\_M 11：PC8\_M |
| FalubE[1:0] | O | 输出到E级ALUb的转发信号 00：V2\_E 01：M\_Wdata 10：AO\_M 11：PC8\_M |
| FPCF [2:0] | O | 输出到F级PC的转发信号 000：RF.V1 001：M\_Wdata 010：AO\_M 011：PC8\_M 100: PC8\_E |
| FdmdataM | O | 输出到M级DMin的转发信号 0：V2\_M 1：M\_Wdata |

1. 功能定义

表格 30hazard功能定义

|  |  |  |
| --- | --- | --- |
| 序号 | 功能名称 | 功能描述 |
| 1 | 产生暂停信号 | 根据Tuse和Tnew产生暂停信号 |
| 2 | 产生转发信号 | 根据Tuse和Tnew产生转发信号 |

1. Bridge和timer
2. 端口说明

表格 31bridge和timer端口说明

|  |  |  |
| --- | --- | --- |
| 信号名 | 方向 | 描述 |
| clk | I | 时钟信号 |
| reset | I | 复位信号 1：复位 0：无效 |
| ADD\_I[1:0] | I | 写入寄存器信号 0：ctrl 1：preset 2：count |
| WE\_I | I | 写入控制信号 1：写入 0：无效 |
| DAT\_I[31:0] | I | 写入数据 |
| DAT\_O[31:0] | O | 输出数据 |
| IRQ | O | Timer中断信号 1：中断 0：无效 |
| PrAddr [31:0] | I | 识别写入timer0或timer1 |
| timer0we | O | Timer0指示信号 1：对timer0处理 0：无效 |
| Timer1we | O | Timer1指示信号 1：对timer1处理 0：无效 |

1. 功能定义

表格 32bridge和timer功能定义

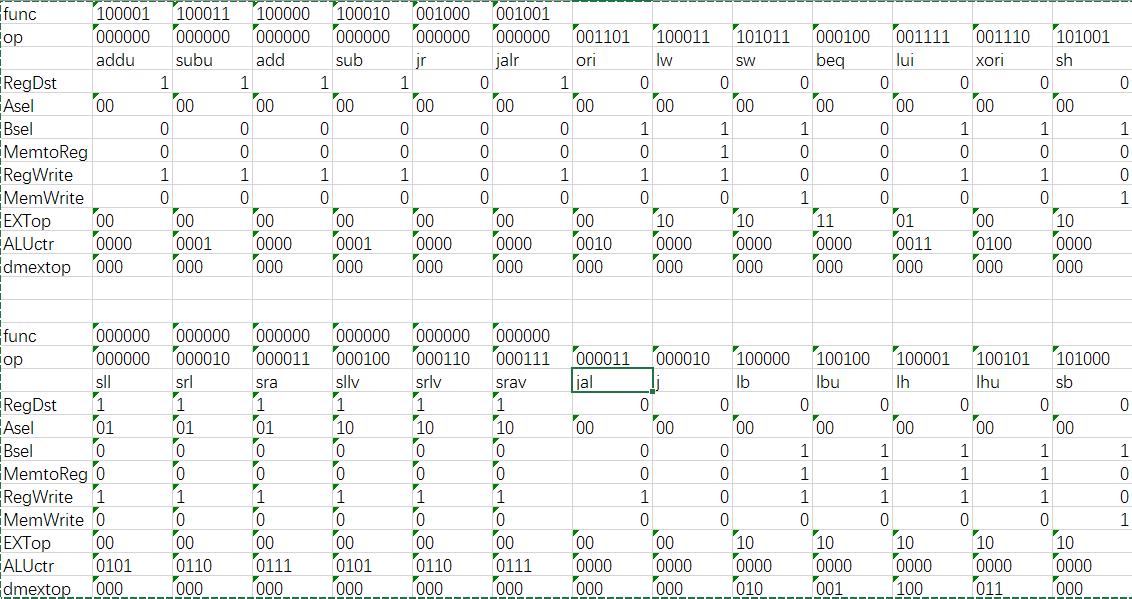
|  |  |  |
| --- | --- | --- |
| 序号 | 功能名称 | 功能描述 |
| 1 | 复位 | 复位信号有效时，HI和LO寄存器和信号被置为0 |
| 2 | 产生中断信号 | 倒数至0时产生中断信号 |
| 3 | 读取数据 | 读取timer中数据 |
| 4 | 写入数据 | 向timer中写入数据 |

1. 控制器设计
2. 端口说明

表格 33ctr端口说明

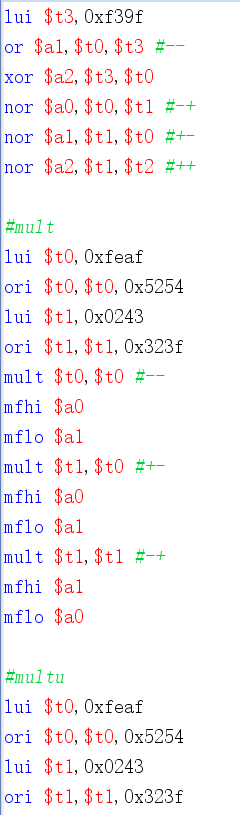
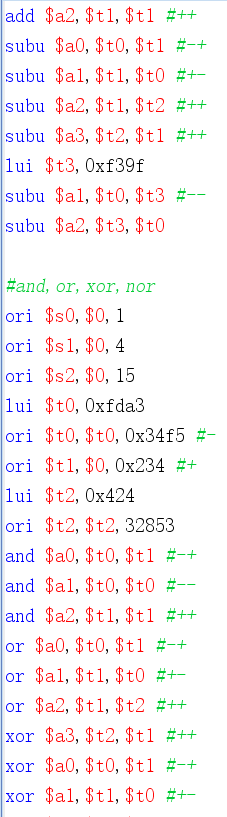
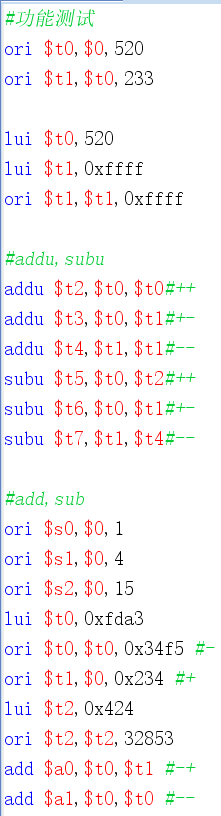
|  |  |  |
| --- | --- | --- |
| 信号名 | 方向 | 描述 |
| instruction[31:0] | I | 32位指令 |
| WR\_sel | O | grf写寄存器决定信号 0：rd 1：rt 2：31号寄存器 |
| A\_sel | O | alu输入数据A决定信号 0：MFaluaE 1：IR[10:6] 2：MFaluaE[4:0] |
| B\_sel | O | alu输入数据B决定信号 1：32位立即数 0：GRF寄存器2输出值 |
| RegWrite | O | grf写寄存器信号 1：写寄存器 0：无效 |
| MemWrite | O | 写内存dm信号 1：写入内存 0：无效 |
| PC\_sel | O | PC跳转信号`ADD4：跳转PC+4 `NPC：跳转npc `RFV1：跳转寄存器堆1 |
| EXTop[1:0] | O | 扩展控制信号 00：无符号扩展 01：低16位补0 10：有符号扩展 11：有符号扩展后逻辑左移两位 |
| ALUctr[3:0] | O | alu控制信号0000加 0001减 0010或 0011B 0100异或 0101逻辑左移 0110逻辑右移 0111算数右移 1000与 1001或非 1010有符号小于置1 1011无符号小于置1 |
| beq | O | Beq指示信号 0：无效 1：beq |
| bne | O | bne指示信号 0：无效 1：bne |
| Bgez | O | Bgez指示信号 0：无效 1：Bgez |
| Blez | O | Blez指示信号 0：无效 1：Blez |
| Bgtz | O | Bgtz指示信号 0：无效 1：Bgtz |
| blez | O | blez指示信号 0：无效 1：blez |
| Muldivop[2:0] | O | 乘除模块控制信号 0：无效 1：mult 2：multu 3：div 4：divu 5：mthi 6：mtlo |
| WDsel[1:0] | O | 写入寄存器堆数据选择信号 0：DR\_W 1：PC8\_W 2：AO\_W |
| Dmexto[2:0] | O | Dm扩展器控制信号 0：无效 1：lbu 2：lb 3：lhu 4：lh |

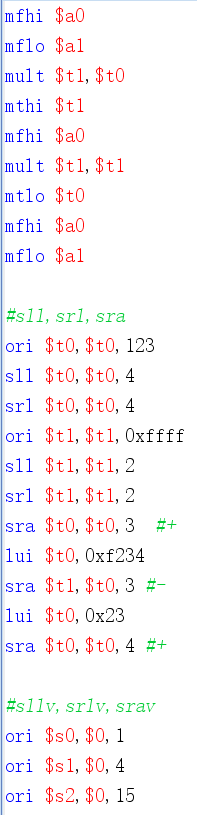
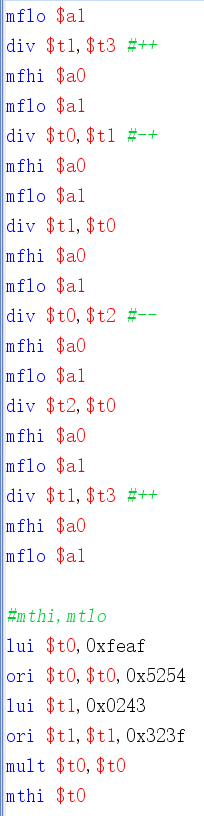
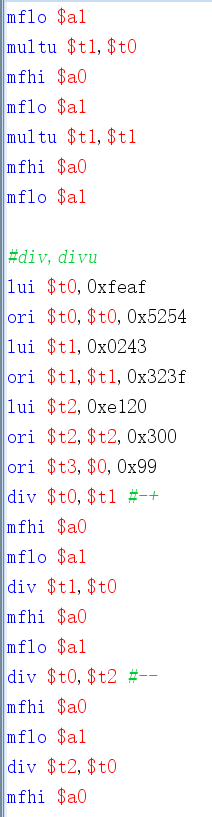
1. 真值表

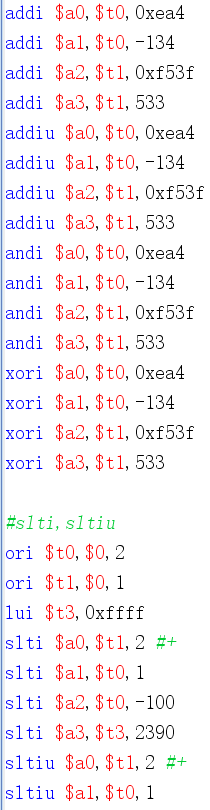
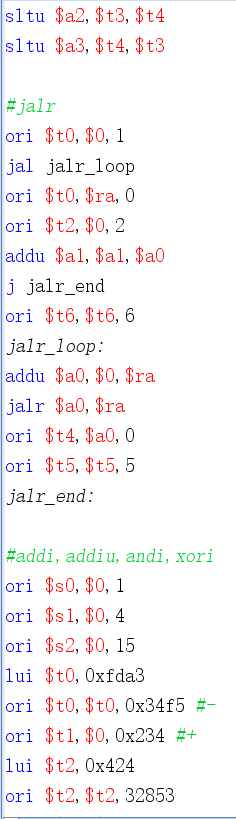
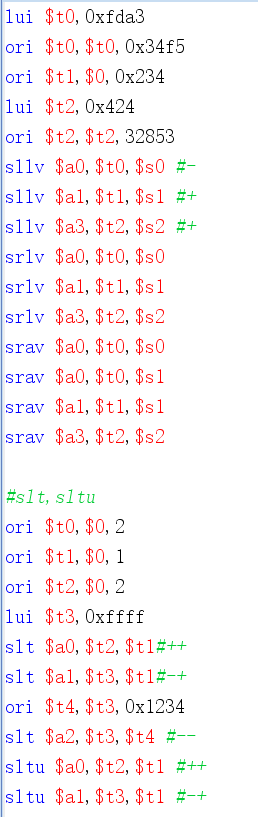
表格 34ctr真值表

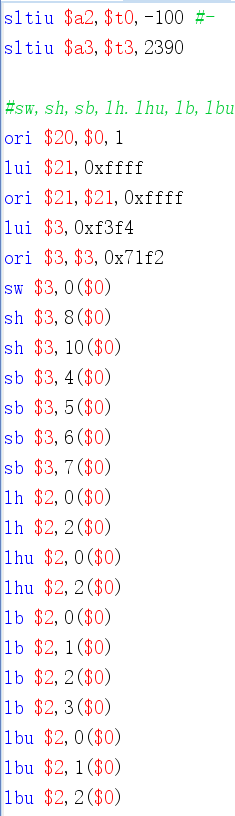
1. 测试
2. 测试程序

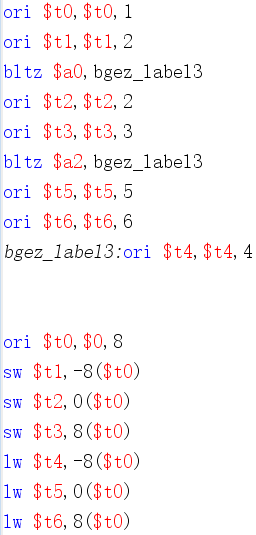
功能测试：



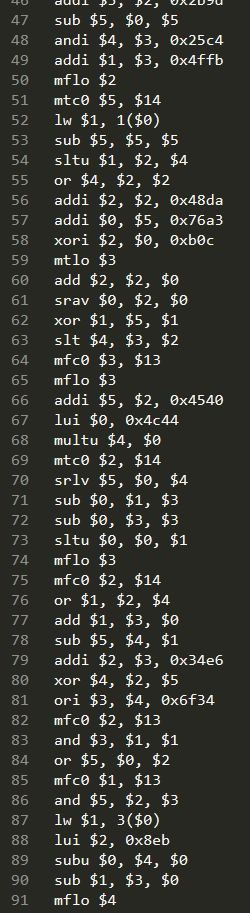
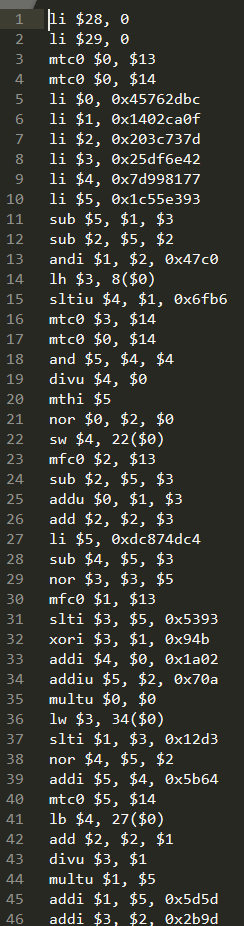


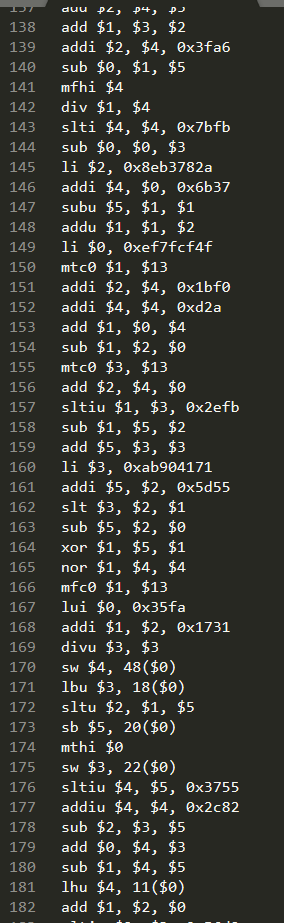
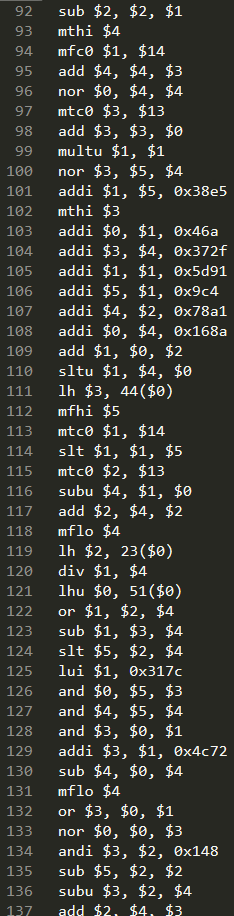


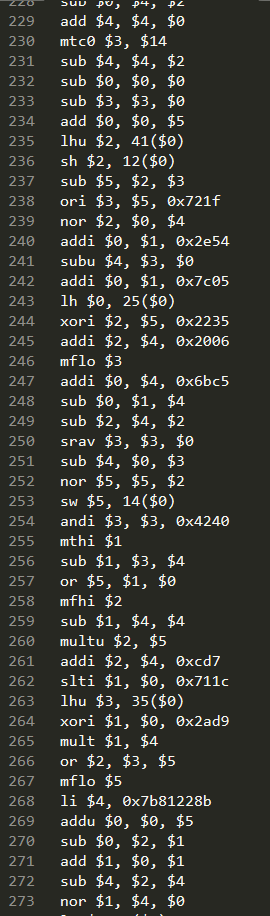
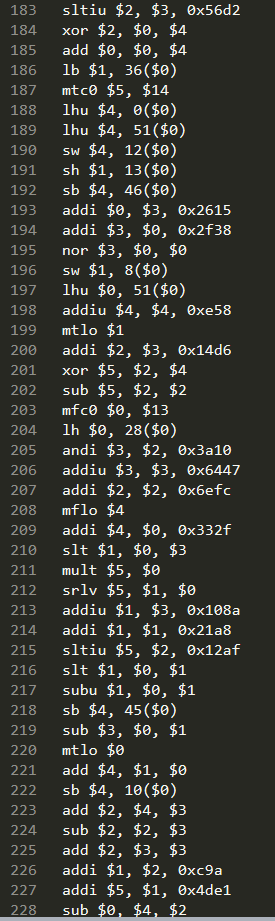


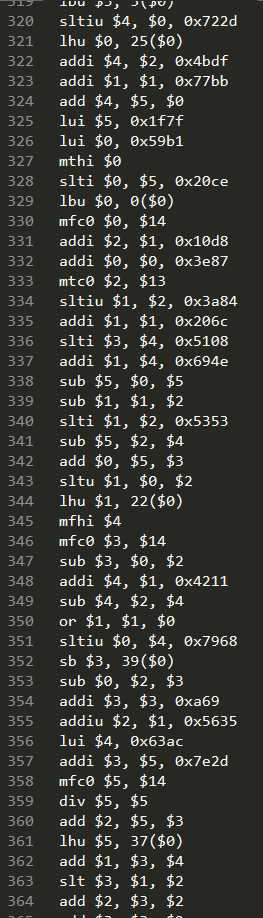
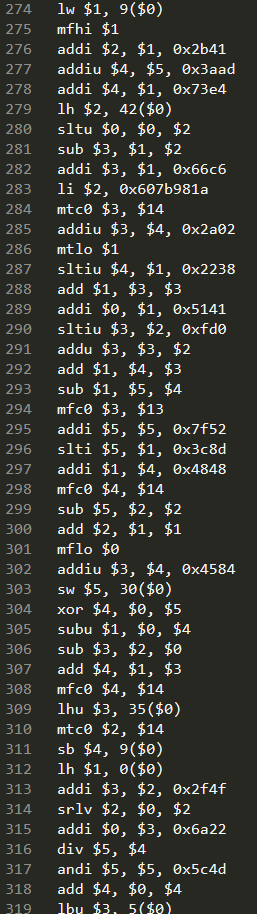


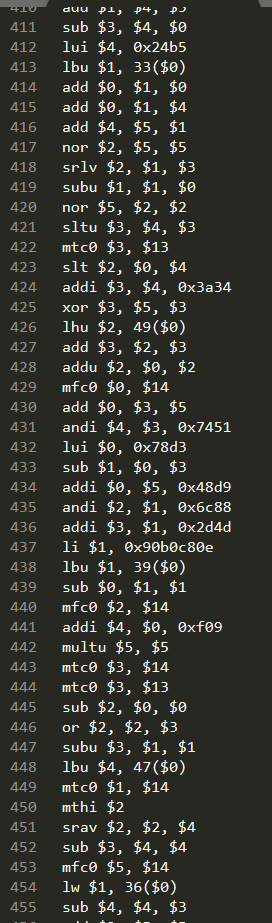
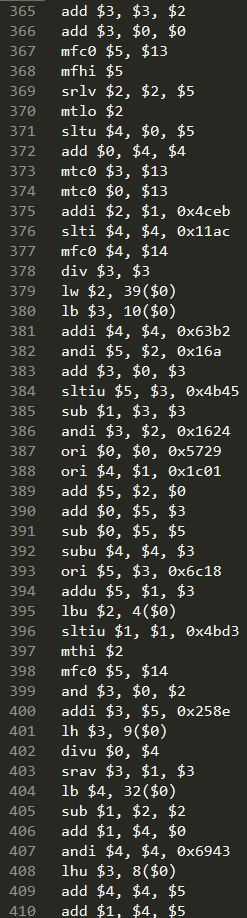
异常测试：

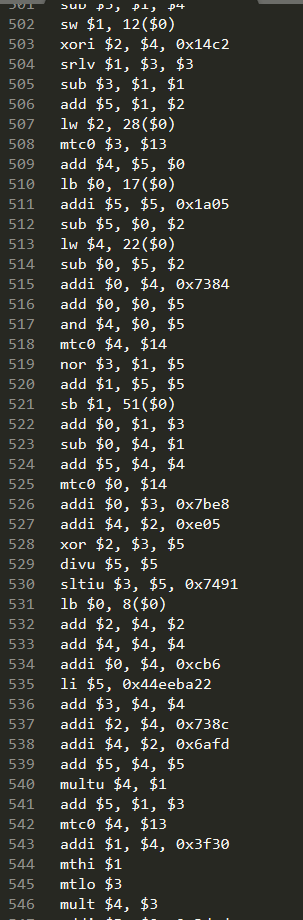
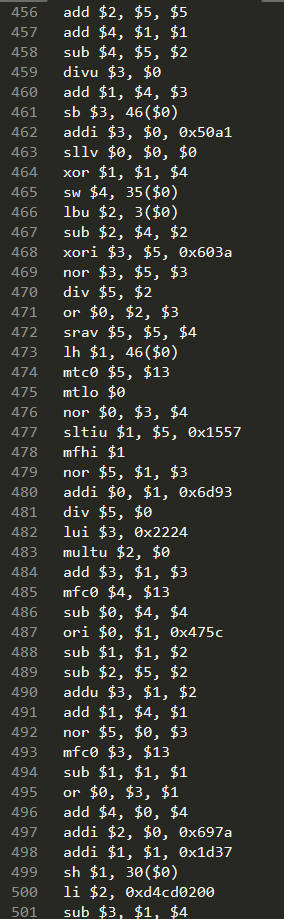


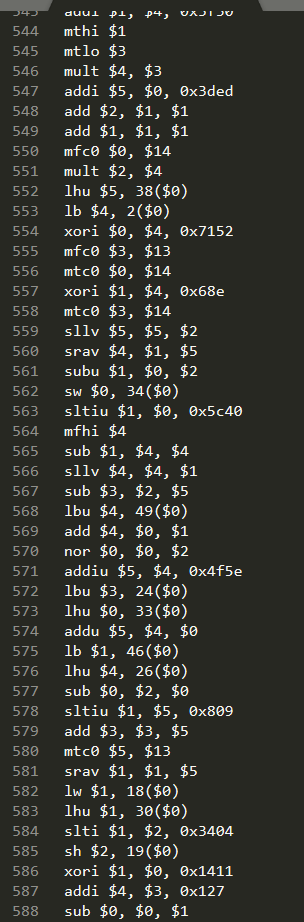












1. 结论

期望输出与实际输出相同。

1. 思考题
2. 我们计组课程一本参考书目标题中有“硬件/软件接口”接口字样，那么到底什么是“硬件/软件接口”？

硬件/软件接口：在接口一端是中断处理程序和用于不同设备的设备驱动程序，另一端是各种硬件设备的控制器，如键盘控制器、打印机控制器。由于设备种类繁多，故该接口相当复杂。

1. 在我们设计的流水线中，DM 处于 CPU 内部，请你考虑现代计算机中它的位置应该在何处。

计算机中的存储器按用途存储器可分为主存储器（内存）和辅助存储器（外存）。外存通常是磁性介质或光盘等，能长期保存信息，容量巨大。内存（即主存）指主板上的存储部件，用来存放当前正在执行的数据和程序，容量大、速度较慢。

无论是IM指令存储器还是DM数据存储器，以及所有设备，都应该位于CPU外部，因为DM相当于现代计算机中的外存，现代计算机通过总线来传输所有的数据和信号，达到沟通的目的。

1. BE 部件对所有的外设都是必要的吗?

并不是对所有外设都需要。比如本实验用的两个timer只需要支持sw，而不需要sb、sh，而对于需要支持按半字和字节写入的外设，比如dm来说，BE是必须的。BE是否使用，取决于外设的需求。

1. 请开发一个主程序以及定时器的exception handler。整个系统完成如下功能。

定时器在主程序中被初始化为模式

定时器倒计数至0产生中断

handler设置使能Enable为 1从而再次启动定时器的计数。2及 3被无限重复。设定初值寄存器的为某个值，如100 或 1000。（注意，主程序可能需要涉及对CP0.SR的编程，推荐阅读过后文再进行。

ori $t4, $0, 64513

mtc0 $t4, $12

ori $t1, $0, 0

ori $t1, $t1, 0x7f00

ori $t2, $0, 0

ori $t2, $t2, 1000#preset

sw $t2, 4($t1) #preset存入相关寄存器

ori $t2, $0, 0

ori $t2, $t2, 9

sw $t2, ($t1)#启动倒数

ori $t3, $0, 1

ori $t3, $0, 2

ori $t3, $0, 3

ori $t3, $0, 4

ori $t3, $0, 5

ori $t3, $0, 6

ori $t3, $0, 7

ori $t3, $0, 8

ori $t3, $0, 9

ori $t3, $0, 10

ori $t3, $0, 1

ori $t3, $0, 2

ori $t3, $0, 3

ori $t3, $0, 4

ori $t3, $0, 5

ori $t3, $0, 6

ori $t3, $0, 7

ori $t3, $0, 8

ori $t3, $0, 9

ori $t3, $0, 10

1. 请查阅相关资料，说明鼠标和键盘的输入信号是如何被CPU知晓的？

键盘、鼠标这类的低速设备是通过中断请求的方式进行IO操作的。即当键盘上按下一个按键的时候，键盘会发出一个中断信号，中断信号经过中断控制器传到CPU，然后CPU根据不同的中断号执行不同的中断响应程序，然后进行相应的IO操作，把按下的按键编码读到寄存器（或者鼠标的操作），最后放入内存中。