|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test** | **Conditions** | **Procedures** | **Expected Results** | **Actual Results** |
|  | | | | |
| Functionality of “Reset” | Code compiled and loaded into ModelSIm | Reset set to 0 (active low);  Push Key[0] button on DE2 board | CurrState = Reset;  CurrPC = 0x0080;  CurrOp = sethi;  All register values and inputs set to 0000 | CurrState = Reset;  CurrPC = 0x0080;  CurrOp = sethi;  All register values and inputs set to 0000 |
| Functionality of “Run” | Code compiled and loaded into ModelSIm | Wait 50 ns after Reset | Run = 1;  Reset = 0;  NextState = Fetch | Run = 1;  Reset = 0;  NextState = Fetch |
| Load input value onto Register 1 (sethi operation) | Code compiled and loaded into ModelSIm; input detected from RegDataIn | Run on high, Reset on high;  sethi operation on R1 with 0x0100 | RegDataIn = 0x0100;  R1out = 0x0100;  CurrPC = 0x0081 | RegDataIn = 0x0100;  R1out = 0x0100;  CurrPC = 0x0081 |
| Load input value onto Register 2 (sethi operation) | Code compiled and loaded into ModelSIm; input detected from RegDataIn | Run on high, Reset on high;  sethi operation on R2 with 0x1800 | RegDataIn = 0x1800;  R2out = 0x1800;  CurrPC = 0x0082 | RegDataIn = 0x1800;  R2out = 0x1800;  CurrPC = 0x0082 |
| Load value onto R3 via addx between R0, R2 | Code compiled and loaded into ModelSIm; Register 0 loaded with all zeroes; value in Register 2 | Run on high, Reset on high;  mem\_rd switched to active high to access R2; | R0out = 0x0000;  R2out = 0x1800;  R3out = 0x1800 | R0out = 0x0000;  R2out = 0x1800;  R3out = 0x1800 |
| Add 1 to register value (addi) | Code compiled and loaded into ModelSIm | Run on high, Reset on high;  Load 1 into const value | Const = 0x0001  R3out = 0x1801 | Const = 0x0001  R3out = 0x1801 |
| Load value onto R4 via sllx on the value in R2 | Code compiled and loaded into ModelSIm; value loaded onto R2 | Run on high, Reset on high; | R4out = 3000 | R4out = 3000 |
| Store register value into memory address (st operation) | Code compiled and loaded into ModelSIm | Run on high, Reset on high;  Switch mem\_wr to active high  Load address to address bus | Addr\_bus = 0x3000  M[0x3000] <= 0x1801 | Addr\_bus = 0x3000  M[0x3000] <= 0x1801 |
| Load value from memory and put it into a register (ld operation  Ld R6,[R5]) | Code compiled and loaded into ModelSIm | Run on high, Reset on high;  Mem\_rd on high;  Get address of R5 | Data\_bus <= 0x000F (comes from ROM)  R6out <= 0x000F | Data\_bus <= 0x000F (comes from ROM)  R6out <= 0x000F |
| orx R6, R2, R6 | Code compiled and loaded into ModelSIm | Read values from R2 and R6  Perform orx operation and store result in Register 6 | R6 <=0x180F | R6 <=0x180F |
| Subtract values between two registers  (subx R7, R2, R3) | Code compiled and loaded into ModelSIm | Run on high, Reset on high;  Read values from R2 and R3;  Perform operation and store in R7 | R7 <= 0xFFFF | R7 <= 0xFFFF |
| Branch if negative | Code compiled and loaded into ModelSIm | Read last value used (R7);  If value is negative, go to “dest1”;  At dest1 executes branch always to dest2;  dest2 performs notx R5, R3, followed by nop | Data\_bus <= 0x4401;  Goes to dest1;  Goes to dest2;  R5 <= 0xE7FE  nextOp <= nop | Data\_bus <= 0x4401;  Goes to dest1;  Goes to dest2;  R5 <= 0xE7FE  nextOp <= nop |
| Jmp R1 | Code compiled and loaded into ModelSIm | Program Counter is assigned the value at R1;  Operation followed by nop | PC <= 0x0100;  nextOp <= nop | PC <= 0x0100;  nextOp <= nop |
| Zero out a register | Code compiled and loaded into ModelSIm; PC successfully jumped to new origin at 0x0100 | Run on high, Reset on high;  Mem\_rd on high;  Subx R5, R5, R5 | R5out <= 0x0000 | R5out <= 0x0000 |
| Branch if zero | Code compiled and loaded into ModelSIm | Reads last value (R5);  Since R5 <= 0x0000, branch to dest3; | Data\_bus <= 0x81c0 | Data\_bus <= 0x81c0 |
| Call 0x01C0 | Code compiled and loaded into ModelSIm | Updates Program Counter with given address;  Updates the memory location with the Program Counter value;  Decrements Stack Pointer by 1  Updates Stack Pointer with memory address | PC <= 0x01C0;  M[SP] <= PC;  SP <= SP – 1;  SP <= 0x7FFD | PC <= 0x01C0;  M[SP] <= PC;  SP <= SP – 1;  SP <= 0x7FFD |
| DE2 Load | Code compiled and loaded into DE2 Board | Load from Quartus Programmer | Load light turns on | Load light turns on |
| DE2 Reset | Code compiled and loaded into DE2 Board | Press Key0 with sw17 (run) off | All registers 0 | All registers 0 |
| DE2 Run | Code compiled and loaded into DE2 Board | Turn SW17 on | Not all registers 0 | Not all registers 0 |
| Display Reg 0 Contents | Code compiled and loaded into DE2 Board | SW0 down, SW1 down, SW2 down | highz |  |
| Display Reg 1 Contents | Code compiled and loaded into DE2 Board | SW0 down, SW1 down, SW2 up | 0100 | 0100 |
| Display Reg 2 Contents | Code compiled and loaded into DE2 Board | SW0 down, SW1 up, SW2 down | 1800 | 1800 |
| Display Reg 3 Contents | Code compiled and loaded into DE2 Board | SW0 down, SW1 up, SW2 up | E7FE | 0000 |
| Display Reg 4 Contents | Code compiled and loaded into DE2 Board | SW0 up, SW1 down, SW2 down | 3000 | 0085 |
| Display Reg 5 Contents | Code compiled and loaded into DE2 Board | SW0up, SW1 down, SW2 up | 1801 | FFFF |
| Display Reg 6 Contents | Code compiled and loaded into DE2 Board | SW0 up, SW1 up, SW2 down | 180F | 0089 |
| Display Reg 7 Contents | Code compiled and loaded into DE2 Board | SW0 up, SW1 up, SW2 up | FFFF | 0000 |