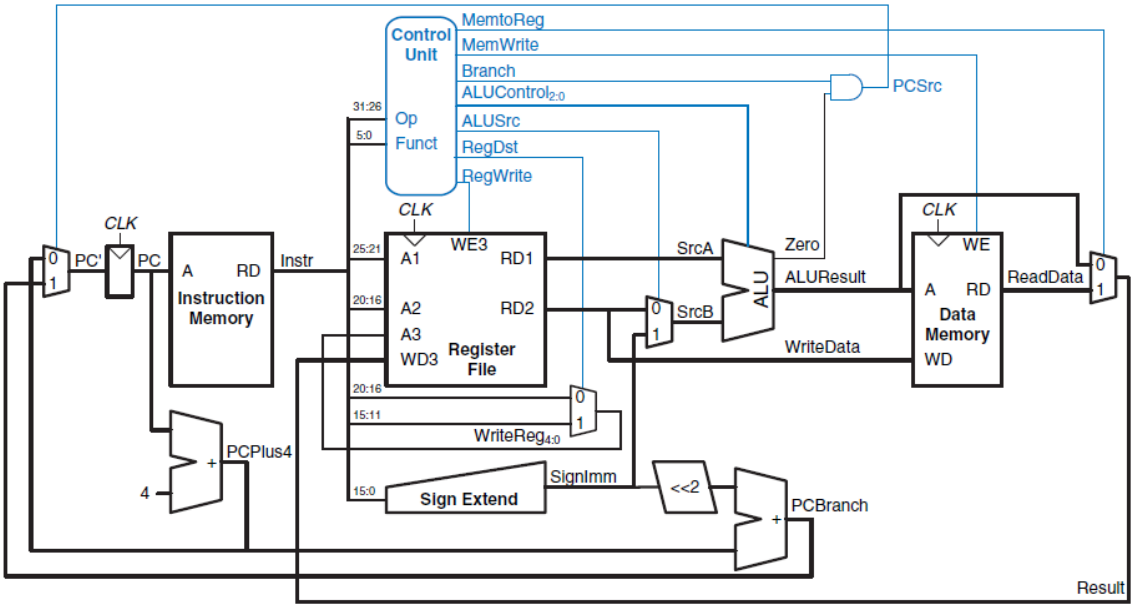
BGTZ -- *Branch on greater than zero :* Branches if the register is greater than zero
if $s > 0 advance_pc (offset << 2)); else advance_pc (4);

opcode name description operation

000111 (7) bgtz rs, label branch if greater than zero if ([rs] > 0) PC = BTA

we don't change in the data-path



Alu :

```
            if((a[31]==1)|(a==0))
                    y=0 ;
            else
                    y=1;
            end

    4'b1011: y = ( b >> (a[4:0]) ); //srlv
    4'b1100: y=(b>>shamt);
    4'b1101: begin      //bgtz
            if(a>0)
                    y=0 ;
            else
                    y=1;
            end
    default: y=0;
    endcase
```

Aludec :

```
module aludec(input logic [5:0] funct,
              input logic [3:0] aluop,
              output logic [3:0] alucontrol);
// assign jr=funct==8;
always_comb

    case(aluop)
        4'b0000: alucontrol <= 4'b0010; // add (for lw/sw/addi)
        4'b0001: alucontrol <= 4'b0110; // sub (for beq)
        4'b0010: alucontrol <= 4'b1010; // blez
        4'b0011: alucontrol <= 4'b0001; // or (for ori)
        4'b0100: alucontrol <= 4'b1000; // shift 16 (LUI)
        4'b0101: alucontrol <= 4'b1001; // xor (for XORI)
        4'b0110: alucontrol <= 4'b0111; // sub (for slti)
        4'b0111: alucontrol <= 4'b0000; // and (for andi)
        4'b1000: alucontrol <= 4'b1101; // bgtz
        default: case(funct) // R-type instructions
```

Maindec:

```
  6'b000000: controls <= 18'b110000000111100000; // RTYPE
  6'b100011: controls <= 18'b100100100000000000; // LW
  6'b101011: controls <= 18'b000101000000000000; // SW
  6'b000100: controls <= 18'b000010000000100000; // BEQ
  6'b001000: controls <= 18'b100100000000000000; // ADDI
  6'b001101: controls <= 18'b101100000001100000; // ORI
  6'b000010: controls <= 18'b000000010000000000; // J
  6'b000101: controls <= 18'b000010000000110000;  // BNQ
  6'b100001: controls <= 18'b100100100000001000; // LH
  6'b100000: controls <= 18'b100100100000001100; // LB
  6'b100100: controls <= 18'b100100000000000010; // lbu
  6'b001100: controls <= 18'b101100000011100000; //andi
  6'b000011: controls <= 18'b100000010000000001; //jal
  6'b001111: controls <= 18'b100100000010000000; // LUI
  6'b001110: controls <= 18'b100100000010100000; // XORI
  6'b000110: controls <= 18'b000010000001000000; // Blez
  6'b001010: controls <= 18'b100100000011000000; //slti
  6'b000111: controls <= 18'b000010000111000000; // bgtz

  default:   controls <= 18'bxxxxxxxxxxxxxxxxxx; // illegal op
```