```
module UnidadedeControle (
input [6:0] Opcode,
input Clock,
output [2:0] AluOp,
output RegDst, MemRead, MemtoReg, MemWrite, ALUSrc,
RegWrite, PCFunct, BEQ, BNE, ControlJump, Halt,
EnableClock, JAL
);


    reg auxRegDst, auxMemRead, auxMemtoReg, auxMemWrite,
    auxALUSrc, auxRegWrite, auxPCFunct, auxBEQ, auxBNE,
    auxControlJump, auxHalt, auxJAL;

    reg [2:0] auxAluOp;

    reg auxEnable;

    always @(*)

    begin

       case(Opcode)
          6'000000: begin // Tipo R
             auxRegWrite <= 1;
             auxPCFunct <= 1;
             auxAluOp <= 3'b000;
             auxMemRead <= 0;
             auxMemWrite <= 0;
             auxMemtoReg <= 0;
             auxALUSrc <= 0;
             auxRegDst <= 0;
             auxBEQ <= 0;
             auxBNE <= 0;
             auxControlJump <= 0;
             auxHalt <= 0;
             auxJAL <= 0;
          end
          6'000001: begin //ADDI
             auxRegWrite <= 1;
             auxPCFunct <= 1;
             auxAluOp <= 3'b001;
             auxMemRead <= 0;
             auxMemWrite <= 0;
             auxMemtoReg <= 0;
             auxALUSrc <= 1;
             auxRegDst <= 0;
             auxBEQ <= 0;
             auxBNE <= 0;
             auxControlJump <= 0;
             auxHalt <= 0;
             auxJAL <= 0;
          end
          6'000010: begin //SUBI
             auxRegWrite <= 1;
             auxPCFunct <= 1;
             auxAluOp <= 3'b001;
             auxMemRead <= 0;
             auxMemWrite <= 0;
             auxMemtoReg <= 0;
             auxALUSrc <= 1;
             auxRegDst <= 0;
             auxBEQ <= 0;
             auxBNE <= 0;
             auxControlJump <= 0;
             auxHalt <= 0;
             auxJAL <= 0;
          end
          6'000011: begin //ANDI
             auxRegWrite <= 1;
             auxPCFunct <= 1;
             auxAluOp <= 3'b001;
             auxMemRead <= 0;
             auxMemWrite <= 0;
             auxMemtoReg <= 0;
             auxALUSrc <= 1;
             auxRegDst <= 0;
```

```
78                  auxBEQ <= 0;
79                  auxBNE <= 0;
80                  auxControlJump <= 0;
81                  auxHalt <= 0;
82                  auxJAL <= 0;
83              end
84              6'000100: begin //ORI
85                  auxRegWrite <= 1;
86                  auxPCFunct <= 1;
87                  auxAluOp <= 3'b001;
88                  auxMemRead <= 0;
89                  auxMemWrite <= 0;
90                  auxMemtoReg <= 0;
91                  auxALUSrc <= 1;
92                  auxRegDst <= 0;
93                  auxBEQ <= 0;
94                  auxBNE <= 0;
95                  auxControlJump <= 0;
96                  auxHalt <= 0;
97                  auxJAL <= 0;
98              end
99              6'000101: begin //LW
100                 auxRegWrite <= 1;
101                 auxPCFunct <= 1;
102                 auxAluOp <= 3'b001;
103                 auxMemRead <= 1;
104                 auxMemWrite <= 0;
105                 auxMemtoReg <= 1;
106                 auxALUSrc <= 1;
107                 auxRegDst <= 0;
108                 auxBEQ <= 0;
109                 auxBNE <= 0;
110                 auxControlJump <= 0;
111                 auxHalt <= 0;
112                 auxJAL <= 0;
113             end
114             6'000110: begin // LWI      !!!!! VER
115                 auxRegWrite <= 1;
116                 auxPCFunct <= 1;
117                 auxAluOp <= 3'b001;
118                 auxMemRead <= 1;
119                 auxMemWrite <= 0;
120                 auxMemtoReg <= 1;
121                 auxALUSrc <= 1;
122                 auxRegDst <= 0;
123                 auxBEQ <= 0;
124                 auxBNE <= 0;
125                 auxControlJump <= 0;
126                 auxHalt <= 0;
127                 auxJAL <= 0;
128             end
129             6'000111: begin // SW
130                 auxRegWrite <= 0;
131                 auxPCFunct <= 1;
132                 auxAluOp <= 3'b001;
133                 auxMemRead <= 0;
134                 auxMemWrite <= 1;
135                 auxMemtoReg <= 0;
136                 auxALUSrc <= 1;
137                 auxRegDst <= 0;
138                 auxBEQ <= 0;
139                 auxBNE <= 0;
140                 auxControlJump <= 0;
141                 auxHalt <= 0;
142                 auxJAL <= 0;
143             end
144             6'001000: begin // J
145                 auxRegWrite <= 0;
146                 auxPCFunct <= 1;
147 //              auxAluOp <= 3'b001; // ???
148                 auxMemRead <= 0;
149                 auxMemWrite <= 0;
150                 auxMemtoReg <= 0;
151                 auxALUSrc <= 0;
152                 auxRegDst <= 0;
153                 auxBEQ <= 0;
154                 auxBNE <= 0;
```

```
155                auxControlJump <= 1;
156                auxHalt <= 0;
157                auxJAL <= 0;
158             end
159                auxRegWrite <= 0;
160                auxPCFunct <= 1;
161    //          auxAluOp <= 3'b001; // ???
162                auxMemRead <= 0;
163                auxMemWrite <= 0;
164                auxMemtoReg <= 0;
165                auxALUSrc <= 0;
166                auxRegDst <= 0;
167                auxBEQ <= 0;
168                auxBNE <= 0;
169                auxControlJump <= 1;
170                auxHalt <= 0;
171                auxJAL <= 0;
172             end
173             6'001010: begin // JAL
174                auxRegWrite <= 0;
175                auxPCFunct <= 1;
176    //          auxAluOp <= 3'b001; // ???
177                auxMemRead <= 0;
178                auxMemWrite <= 0;
179                auxMemtoReg <= 0;
180                auxALUSrc <= 0;
181                auxRegDst <= 1;
182                auxBEQ <= 0;
183                auxBNE <= 0;
184                auxControlJump <= 1;
185                auxHalt <= 0;
186                auxJAL <= 0;
187             end
188             6'001101: begin // SLTI
189                auxRegWrite <= 0;
190                auxPCFunct <= 1;
191                auxAluOp <= 3'b001;
192                auxMemRead <= 0;
193                auxMemWrite <= 1;
194                auxMemtoReg <= 0;
195                auxALUSrc <= 1;
196                auxRegDst <= 0;
197                auxBEQ <= 0;
198                auxBNE <= 0;
199                auxControlJump <= 0;
200                auxHalt <= 0;
201                auxJAL <= 0;
202             end
203             6'001110: begin // BEQ
204                auxRegWrite <= 0;
205                auxPCFunct <= 1;
206                auxAluOp <= 3'b001;
207                auxMemRead <= 0;
208                auxMemWrite <= 1;
209                auxMemtoReg <= 0;
210                auxALUSrc <= 1;
211                auxRegDst <= 0;
212                auxBEQ <= 0;
213                auxBNE <= 0;
214                auxControlJump <= 0;
215                auxHalt <= 0;
216                auxJAL <= 0;
217             end
218             6'001111: begin // BNE
219                auxRegWrite <= 0;
220                auxPCFunct <= 1;
221                auxAluOp <= 3'b001;
222                auxMemRead <= 0;
223                auxMemWrite <= 1;
224                auxMemtoReg <= 0;
225                auxALUSrc <= 1;
226                auxRegDst <= 0;
227                auxBEQ <= 0;
228                auxBNE <= 0;
229                auxControlJump <= 0;
230                auxHalt <= 0;
231                auxJAL <= 0;
```

```
232             end
233             6'010000: begin // Move
234                auxRegWrite <= 0;
235                auxPCFunct <= 1;
236                auxAluOp <= 3'b001;
237                auxMemRead <= 0;
238                auxMemWrite <= 1;
239                auxMemtoReg <= 0;
240                auxALUSrc <= 1;
241                auxRegDst <= 0;
242                auxBEQ <= 0;
243                auxBNE <= 0;
244                auxControlJump <= 0;
245                auxHalt <= 0;
246                auxJAL <= 0;
247             end
248             6'010001: begin // Nop
249                auxRegWrite <= 0;
250                auxPCFunct <= 1;
251                auxAluOp <= 3'b001;
252                auxMemRead <= 0;
253                auxMemWrite <= 1;
254                auxMemtoReg <= 0;
255                auxALUSrc <= 1;
256                auxRegDst <= 0;
257                auxBEQ <= 0;
258                auxBNE <= 0;
259                auxControlJump <= 0;
260                auxHalt <= 0;
261                auxJAL <= 0;
262             end
263             6'111111: begin // Halt
264                auxRegWrite <= 0;
265                auxPCFunct <= 1;
266                auxAluOp <= 3'b001;
267                auxMemRead <= 0;
268                auxMemWrite <= 1;
269                auxMemtoReg <= 0;
270                auxALUSrc <= 1;
271                auxRegDst <= 0;
272                auxBEQ <= 0;
273                auxBNE <= 0;
274                auxControlJump <= 0;
275                auxHalt <= 0;
276                auxJAL <= 0;
277             end
278       end
279
280       assign RegDst = auxRegDst;
281       assign MemRead = auxMemRead;
282       assign MemtoReg = auxMemtoReg;
283       assign MemWrite = auxMemWrite;
284       assign ALUSrc = auxALUSrc;
285       assign RegWrite = auxRegWrite;
286       assign PCFunct = auxPCFunct;
287       assign BEQ = auxBEQ;
288       assign BNE = auxBNE;
289       assign ControlJump = auxControlJump;
290       assign Halt = auxHalt;
291       assign EnableClock = auxEnable;
292       assign JAL = auxJAL;
293
294
295    endmodule
296
```