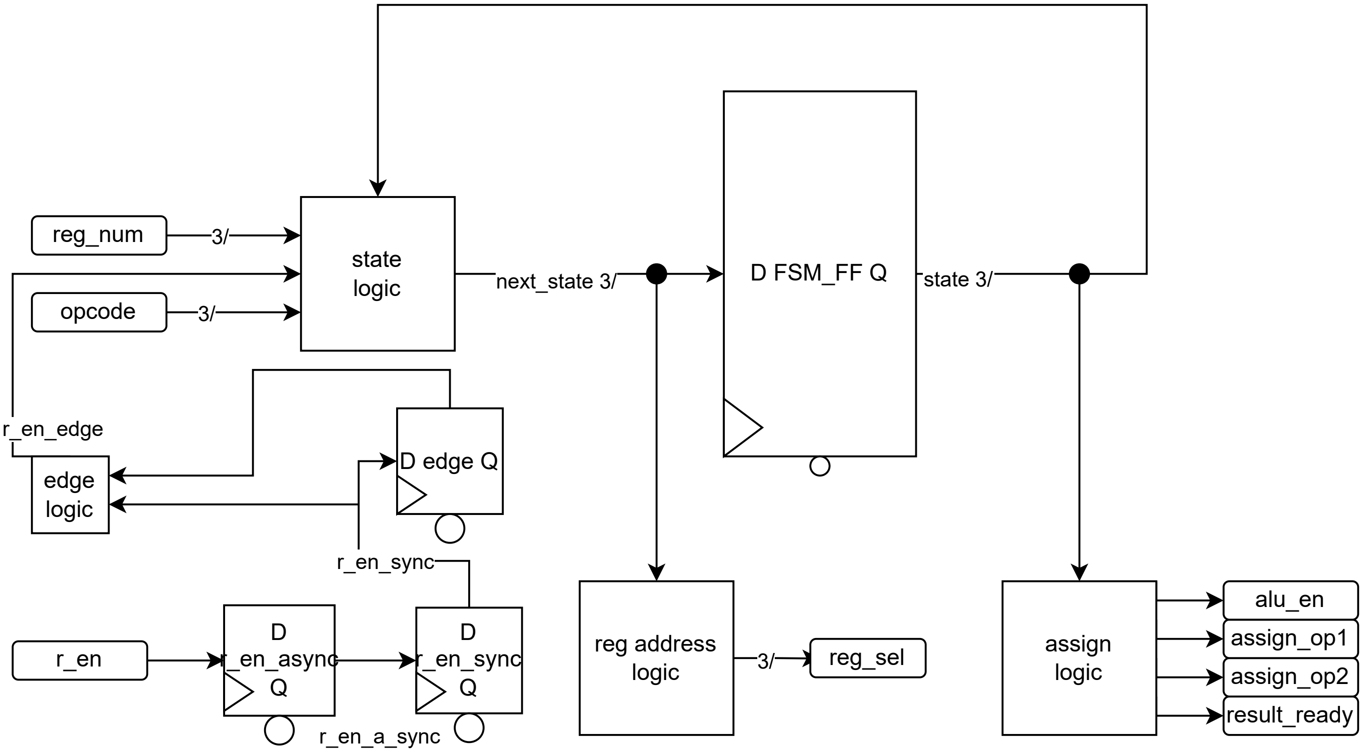

Register Address Logic:

```
reg_sel = 3'b0;
case (next_state)
REG1: reg_sel = reg_num;
REG2: reg_sel = reg_num;
endcase
```

State Logic:

```
next_state = state;
case (state)
IDLE:  if (r_en_edge) next_state = EN;
EN:    if (|reg_num) next_state =
REG1;
REG1:  next_state = IDLE2;
IDLE2: if (|reg_num) next_state =
REG2;
REG2:  next_state = IDLE3;
IDLE3: if (|opcode) next_state =
RESULT;
RESULT: next_state = IDLE;
endcase
```

Assign Logic:

```
alu_en = 1'b0;
assign_op1 = 1'b0;
assign_op2 = 1'b0;
result_ready = 1'b0;
case (state)
  REG1: begin
    assign_op1 = 1'b1;
  end
  REG2: begin
    assign_op2 = 1'b1;
  end

  RESULT : begin
    alu_en = 1'b1;
    result_ready = 1'b1;
  end
endcase
```