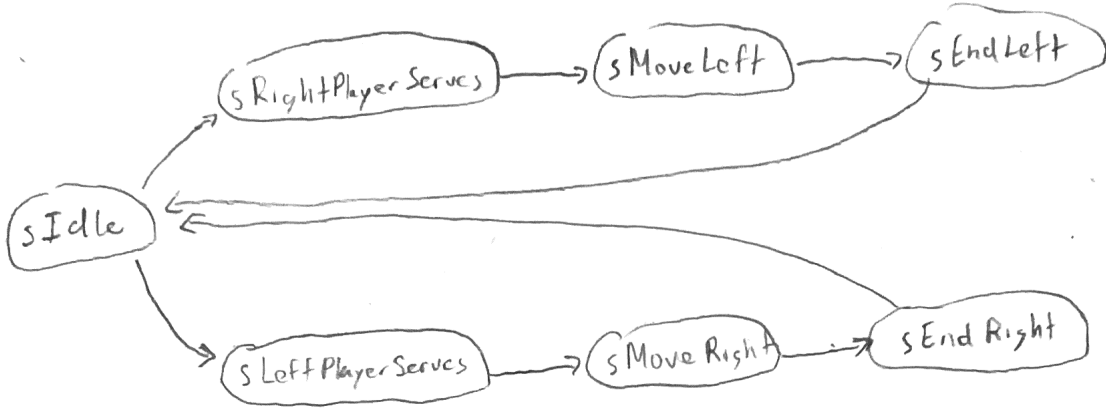
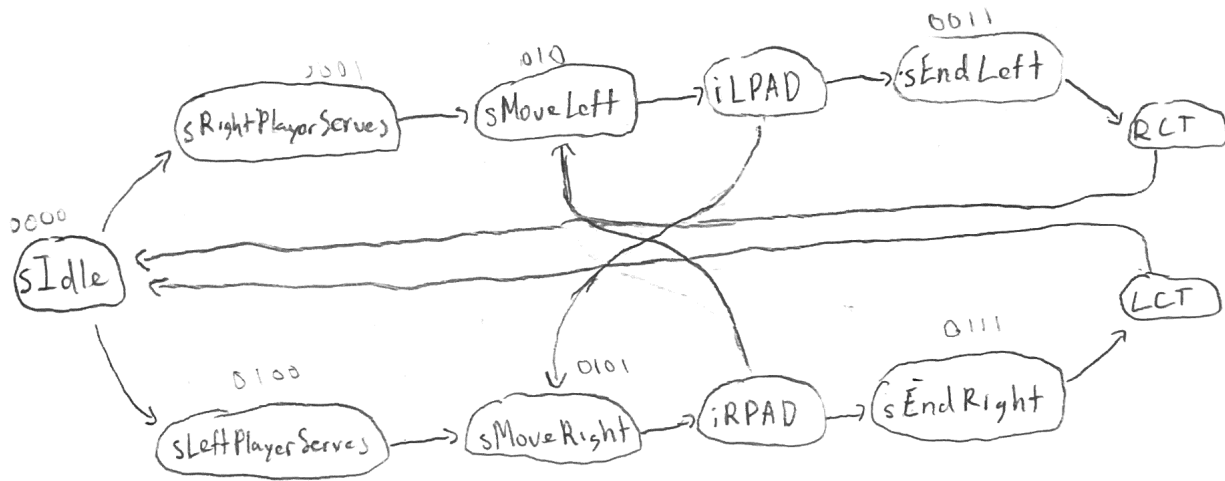


Pre-Lab 9

1a)



1b)



```
2) module pong-controller2 (input QLEFT, QRIGHT, iRESET, iRSRV, iLSRV, iRPAD, iLPAD, clk1,
    output [1:0] S, LCT, RCT, LSI, RSI);
```

```
    reg [2:0] currstate, nextState;
```

```
    parameter sIdle = 3'b000, SRSRV = 3'b001, SMOVL = 3'b010, SENDL = 3'b011,
        SLSRV = 3'b100, SMOVR = 3'b101, SENDR = 3'b111;
```

```
    always @(posedge clk1)
```

```
    always @(*) begin
```

```
        nextState = sIdle;
```

```
        case (currstate)
```

```
            sIdle: begin
```

```
                if (iRSRV) nextState = SRSRV;
```

```
                else if (iLSRV) nextState = SLSRV;
```

```
                {LSI, RSI, LCT, RCT, S} = 6'b000011;
```

```
            end
```

```
            SRSRV: begin
```

```
                if (!iRESET) nextState = SMOVL;
```

```
                {LSI, RSI, LCT, RCT, S} = 6'b100010;
```

```
            end
```

```
            SLSRV: begin
```

```
                if (!iRESET) nextState = SMOVR;
```

```
                {LSI, RSI, LCT, RCT, S} = 6'b010001;
```

```
            end
```

```
            SMOVL: begin
```

```
                if (iRESET) nextState = sIdle;
```

```
                else if (QLeft && iLPAD) nextState = SMOVR;
```

```
                else if (QLeft) nextState = SENDL;
```

```
                else nextState = SMOVL;
```

```
                {LSI, RSI, LCT, RCT, S} = 6'b000010;
```

```
            end
```

```
            SMOVR: begin
```

```
                if (iRESET) nextState = sIdle;
```

```
                else if (QRight && iRPAD) nextState = SMOVL;
```

```
                else if (QRight) nextState = SENDR;
```

```
                else nextState = SMOVR;
```

```
                {LSI, RSI, LCT, RCT, S} = 6'b000001;
```

```
            end
```

```
            SENDL: begin
```

```
                nextState = sIdle;
```

```
                {LSI, RSI, LCT, RCT, S} = 6'b000100;
```

```
            end
```

```
            SENDR: begin
```

```
                nextState = sIdle;
```

```
                {LSI, RSI, LCT, RCT, S} = 6'b001000;
```

```
            end
```

```

        default: begin
            nextState = sIdle;
            { LSI, RSI, LCT, RCT, S } = 6'b000000;
        end
    endcase
end
endmodule

```

3)

Test Procedure

1. Serve from L → R and scores
2. Serve from R → L and scores
3. Test iRESET at serves and movements
4. Check that LPAD sends ball back to right
5. Check that RPAD sends ball back to left
6. Check that Game goes to sIdle after scoring, for both sides.