



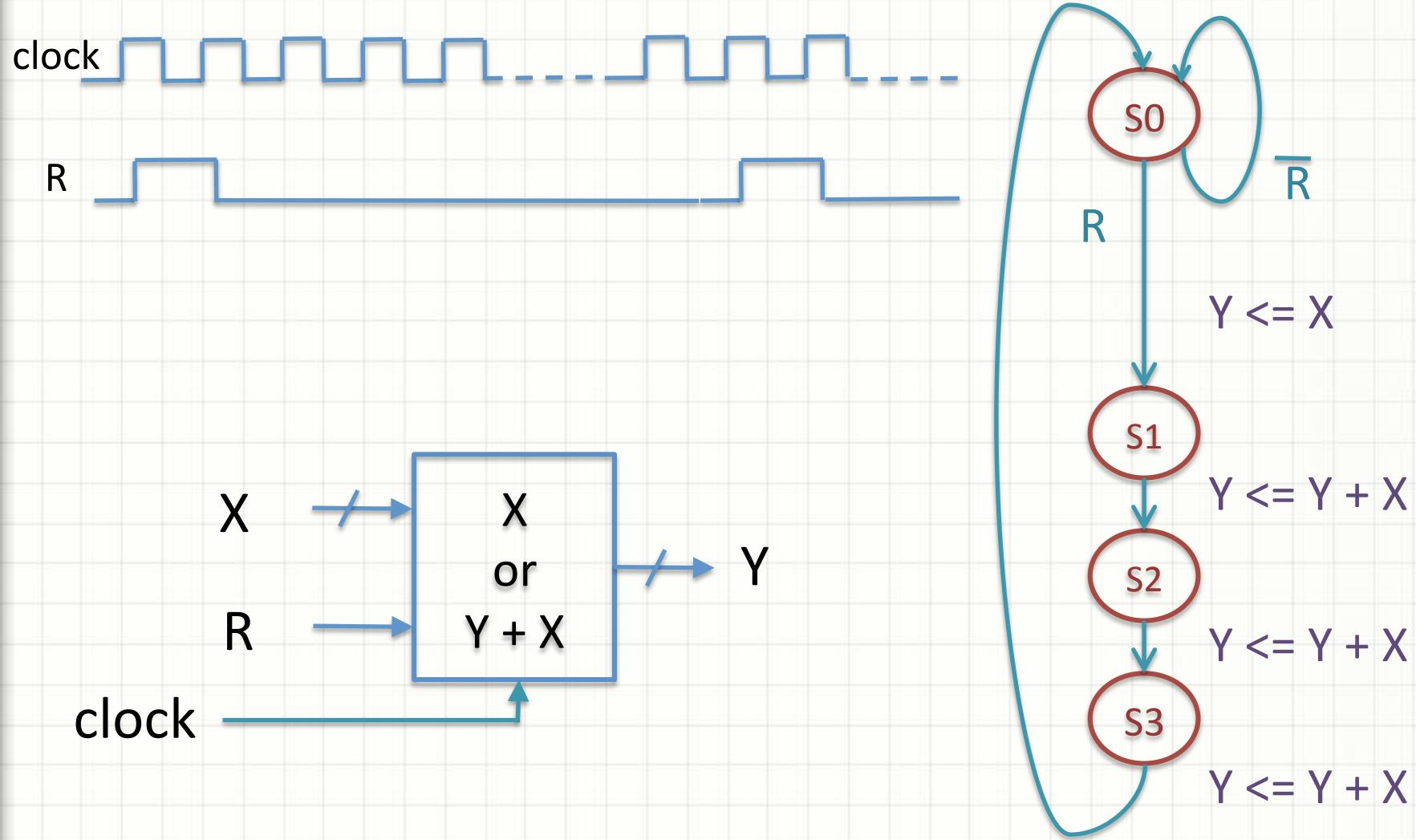
COL215 DIGITAL LOGIC AND SYSTEM DESIGN

System Design
10 October 2017

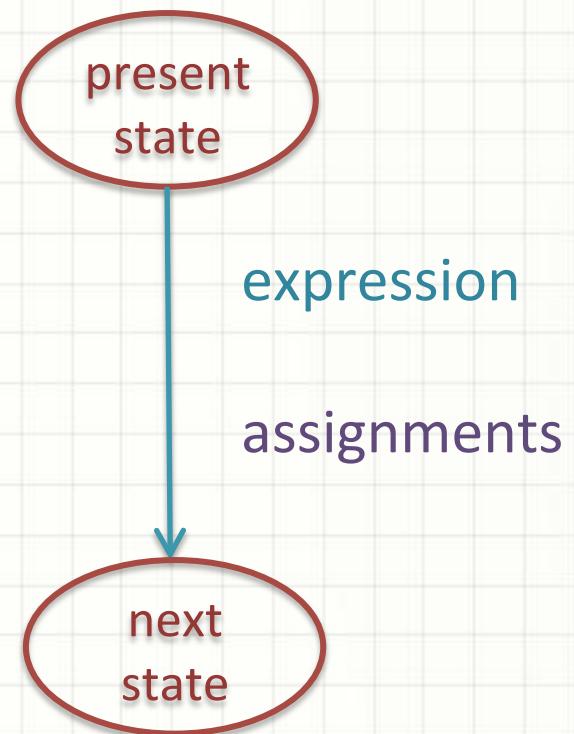
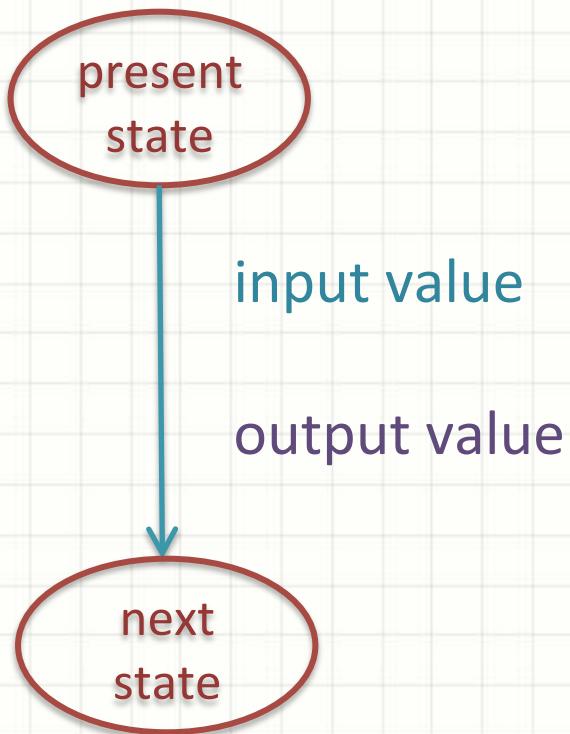
From truth tables and FSMs to Systems

- Abstraction
 - bits => bit vectors, integers
 - boolean operations => vector operations and arithmetic operations
- Hierarchy
 - system, subsystems, modules
- Regularity
 - iterative structures
- Control-Data Partition

Adding 4 numbers sequentially

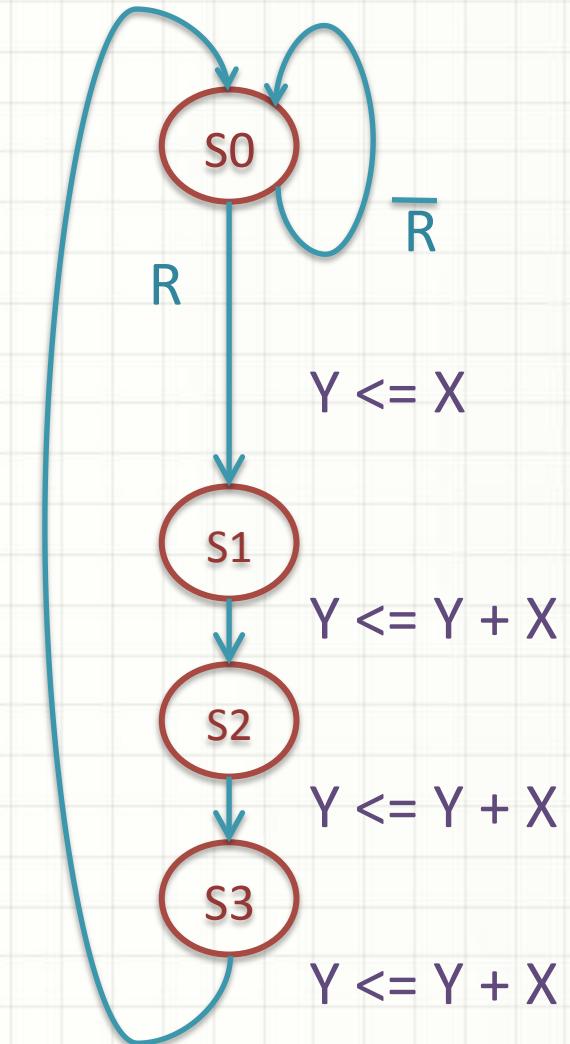
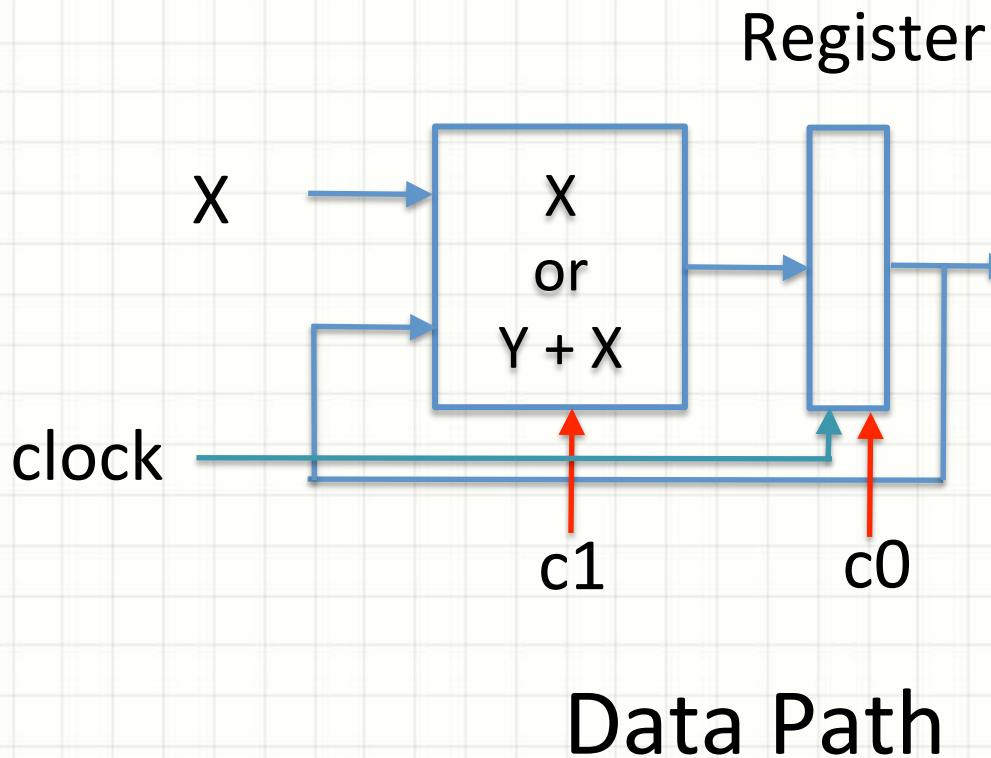


FSM + Data

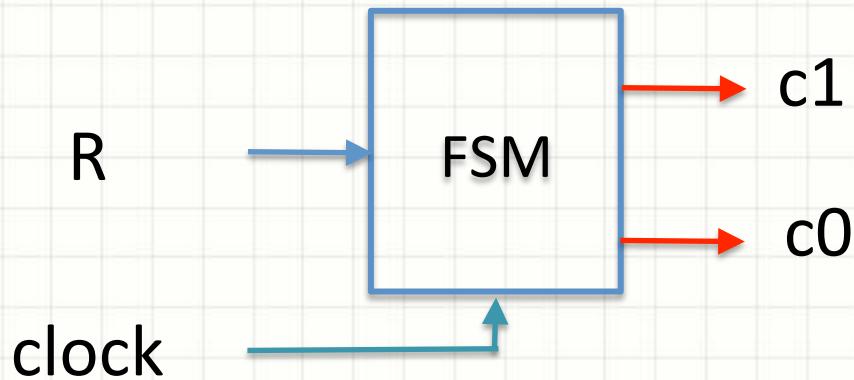


Micro operations / register transfers

- $Y \leq X$
- $Y \leq Y + X$



FSM

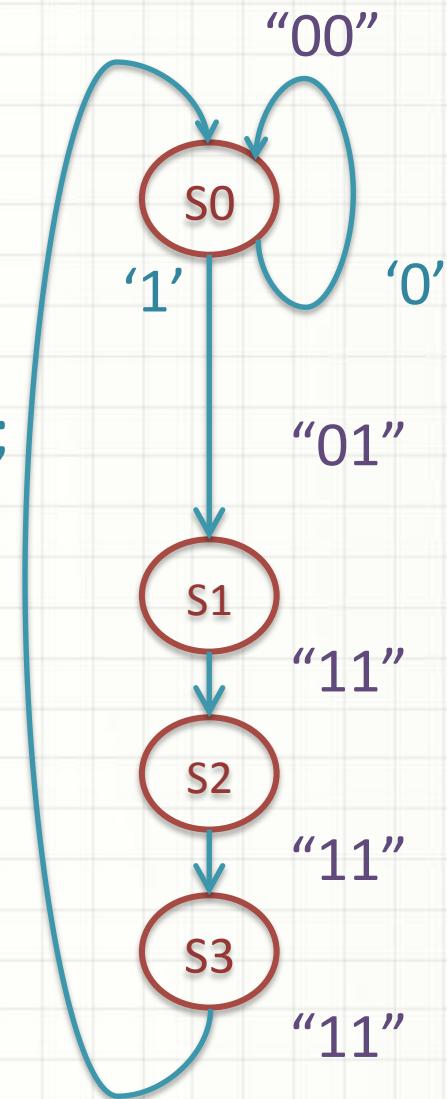


Controller

Controller

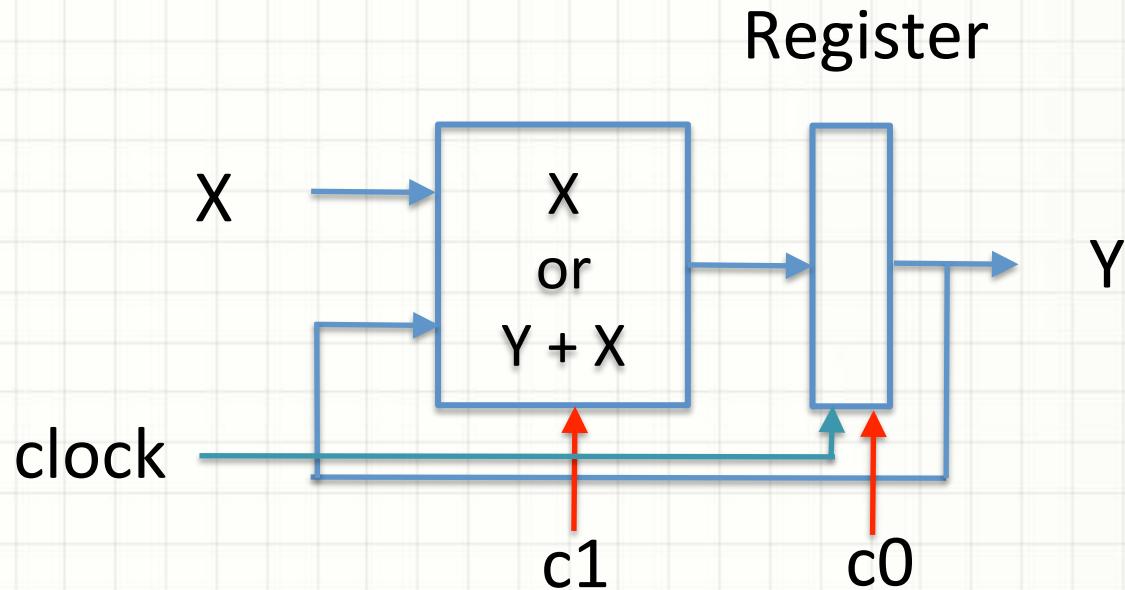
this syntax is valid

```
if rising_edge(clock) then
    case state is
        when S0 =>
            if R = '1' then C <= "01"; state <= S1;
            else C <= "00"; end if;
        when S1 => C <= "11"; state <= S2;
        when S2 => C <= "11"; state <= S3;
        when S3 => C <= "11"; state <= S0;
    end case;
end if;
```

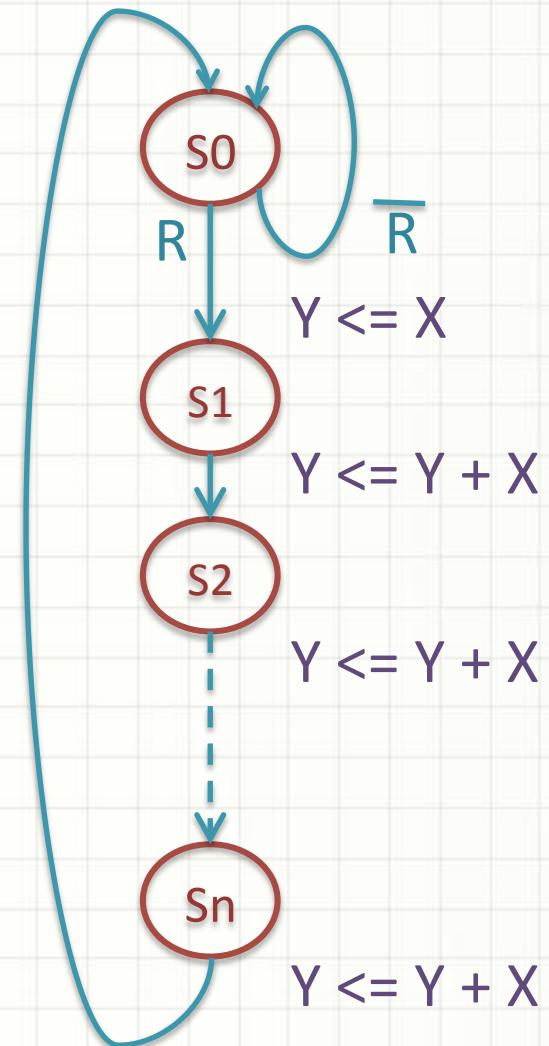
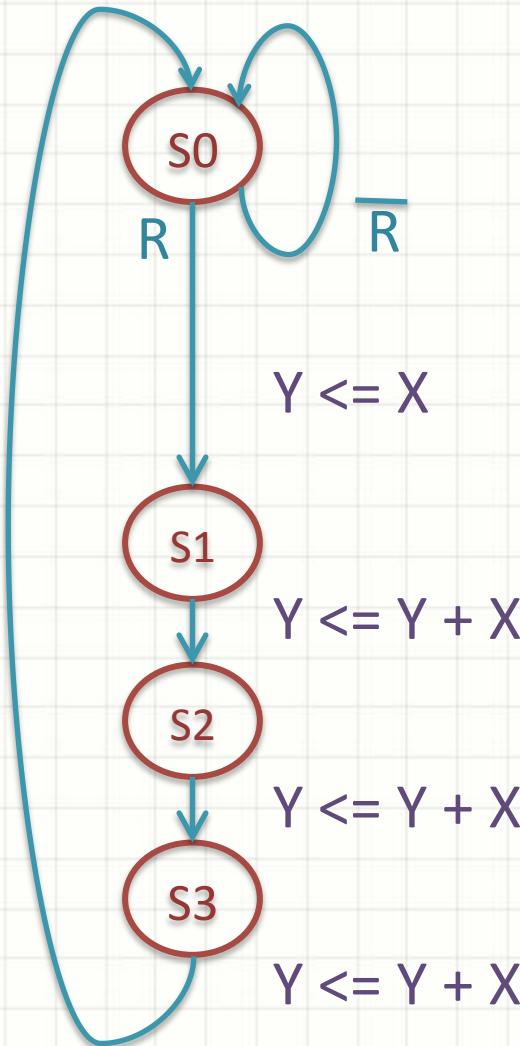


Data path

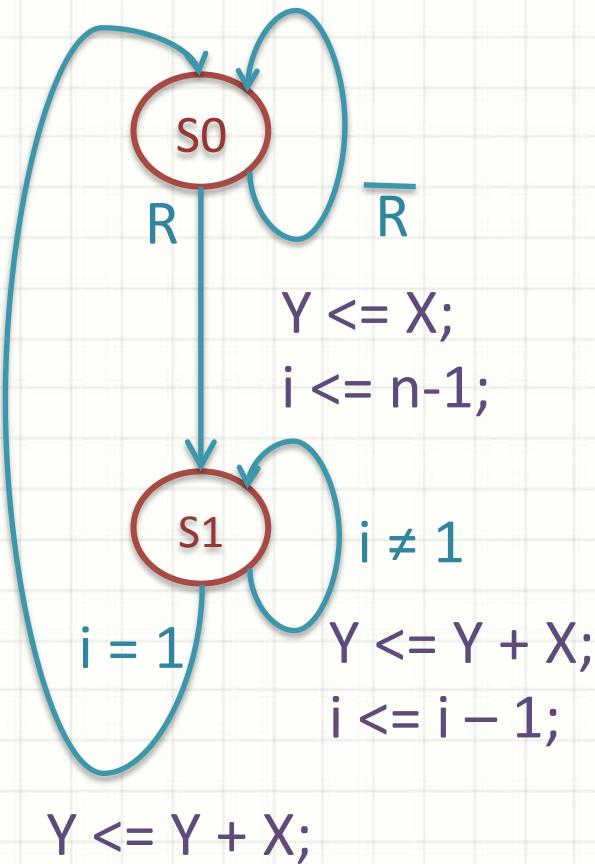
```
if rising_edge(clock) then  
if C(0) = '1' then  
  if C(1) = '1' then Y <= Y + X;  
  else Y <= X;  
end if;  
end if;  
end if;
```



Generalize to addition of n numbers



Generalize to addition of n numbers



Here counting is done by data path, not by controller

Generalize to addition of n numbers

if rising_edge(clock) then

case state is

when S0 =>

if R = '1' then

Y <= X; i <= n-1; state <= S1;

end if;

when S1 =>

Y <= Y + X;

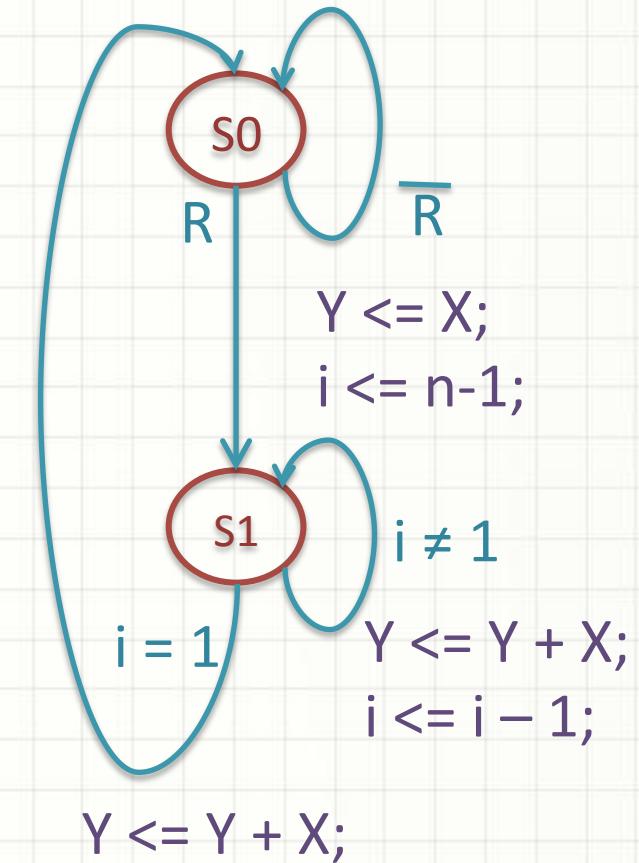
if i = 1 then state <= S0;

else i <= i - 1;

end if;

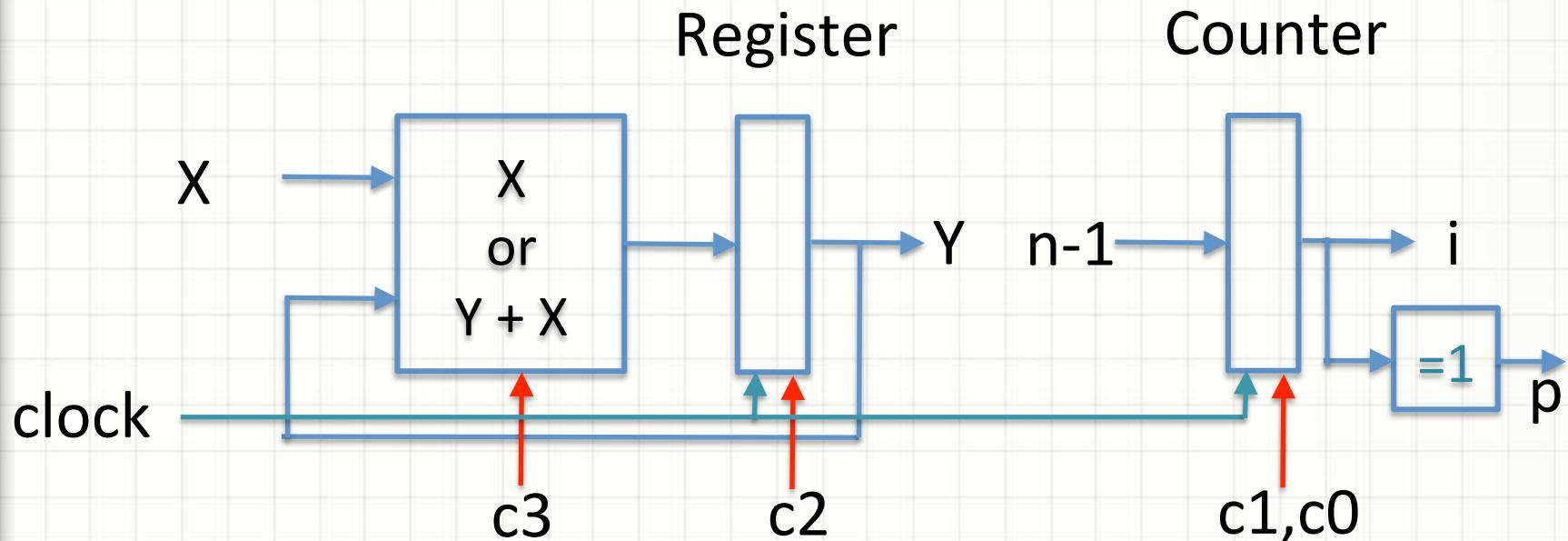
end case;

end if;



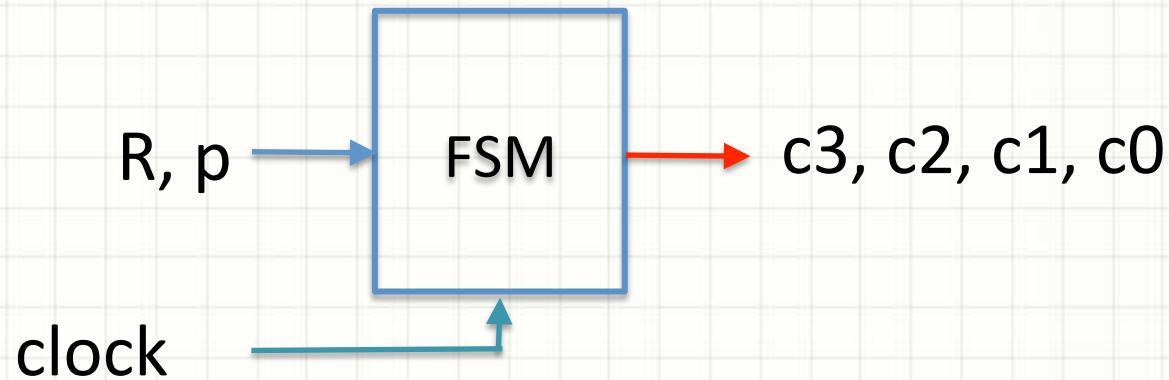
Micro operations / register transfers

- $Y \leq X$ $i \leq n-1$
- $Y \leq Y + X$ $i \leq i - 1$



Data Path

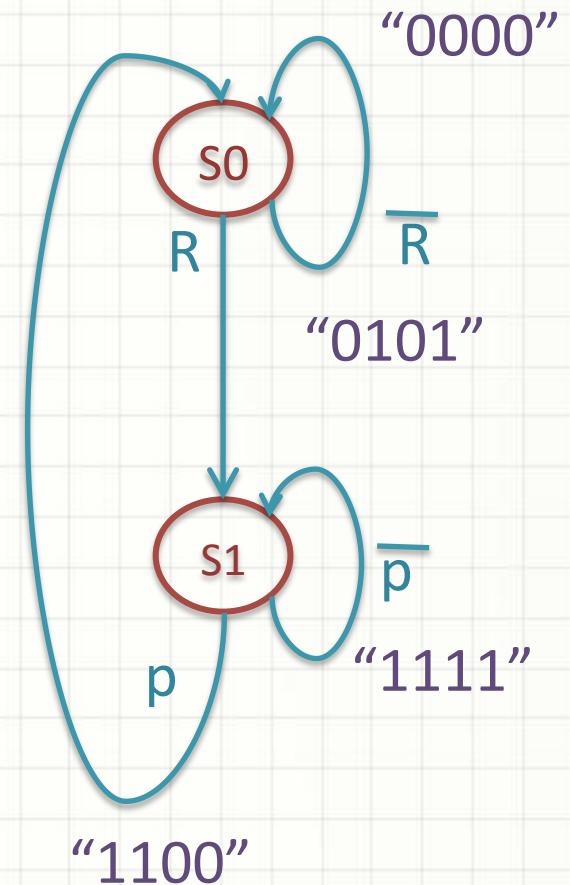
FSM



Controller

Controller

```
if rising_edge(clock) then
    case state is
        when S0 =>
            if R = '1' then
                C <= "0101"; state <= S1;
            else C <= "0000";
            end if;
        when S1 =>
            if p = '1' then
                C <= "1100"; state <= S0;
            else C <= "1111";
            end if;
    end case;
end if;
```



Data path

if rising_edge(clock) then

if C(0) = '1' then

if C(1) = '1' then $i \leq i - 1$; else $i \leq n - 1$; end if;

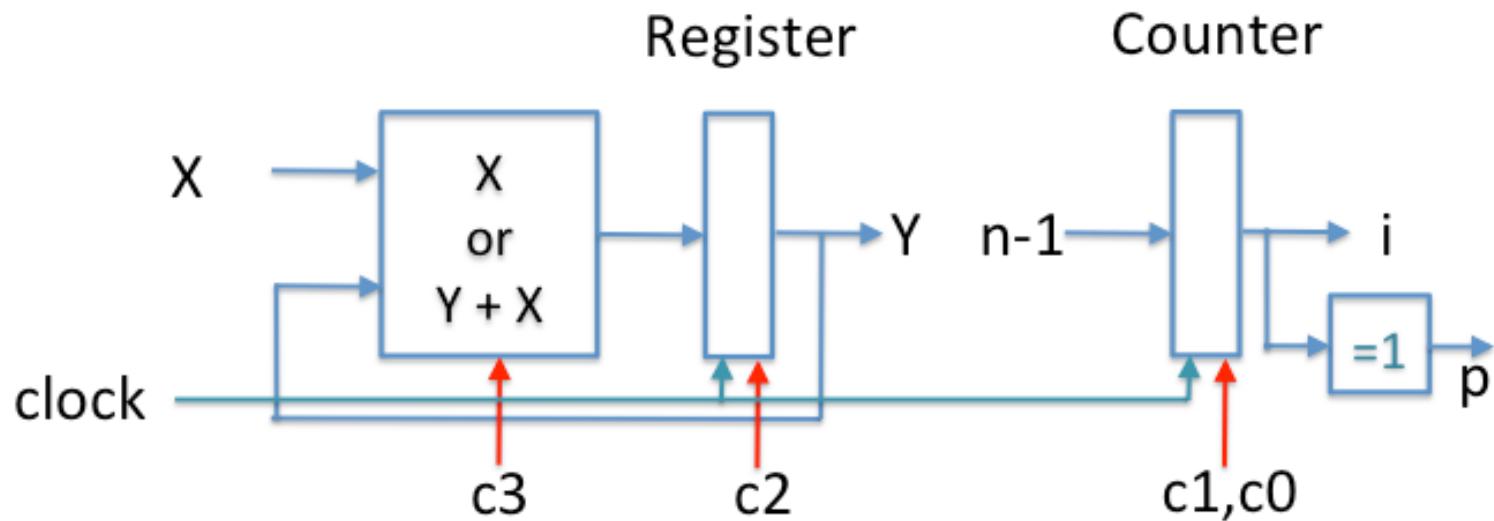
end if;

if C(2) = '1' then

if C(3) = '1' then $Y \leq Y + X$; else $Y \leq X$; end if;

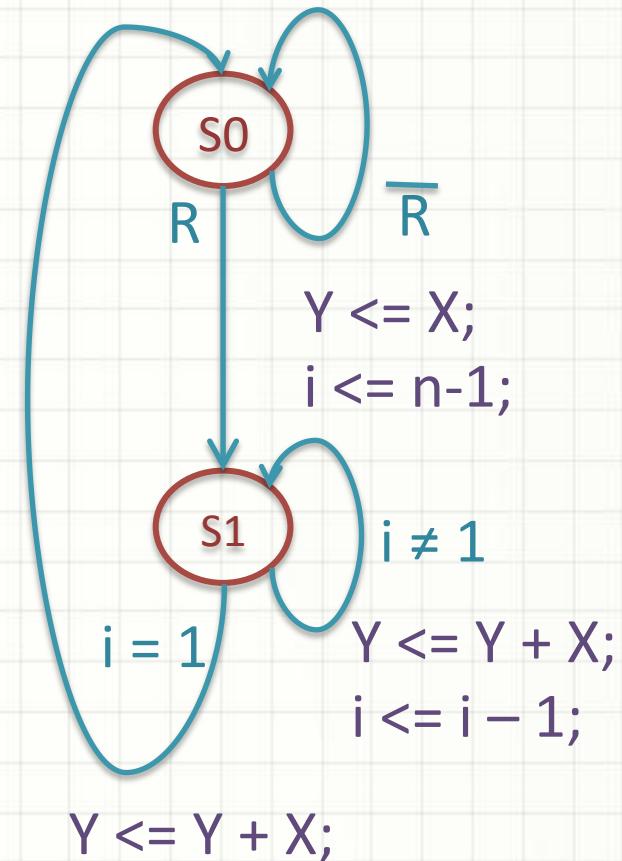
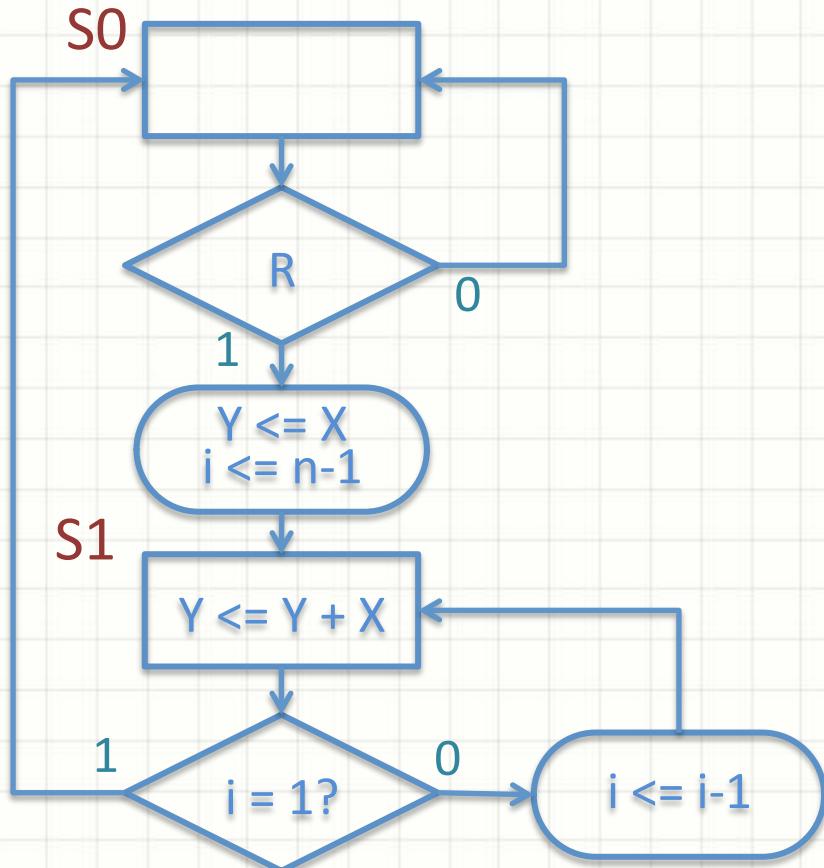
end if;

end if;



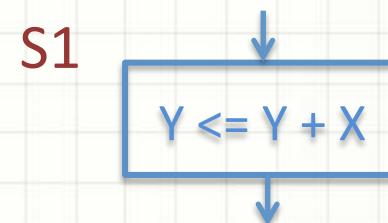
Another notation : ASM charts

ASM = Algorithmic State Machine

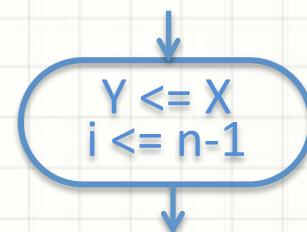


ASM notation

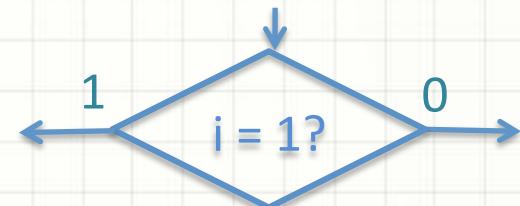
State with actions



Conditional actions

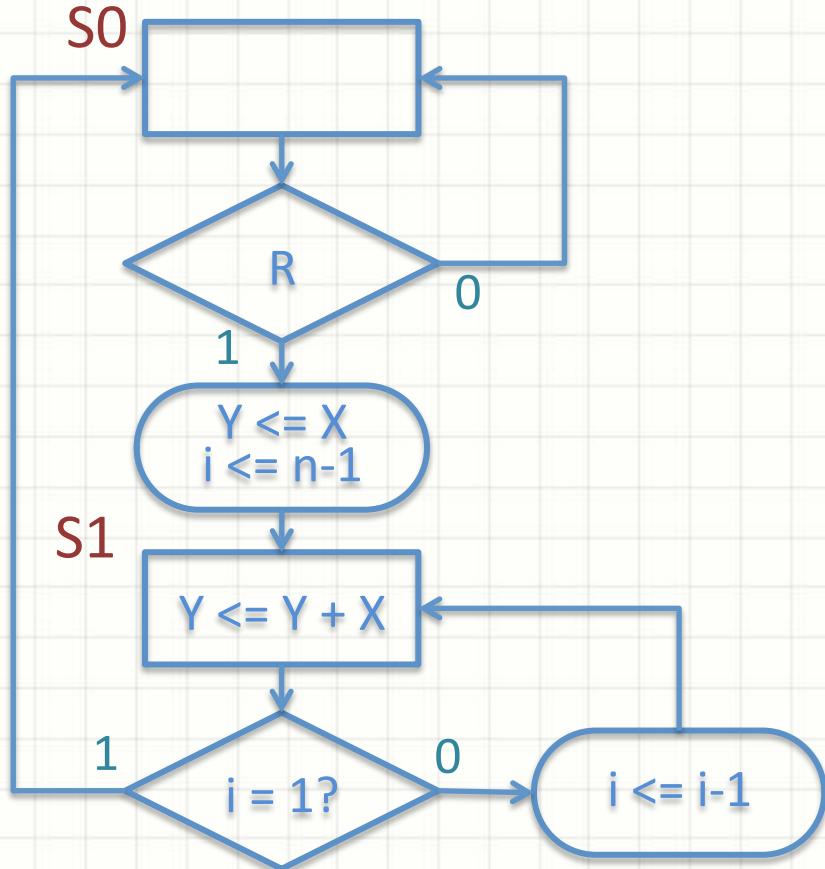


Condition



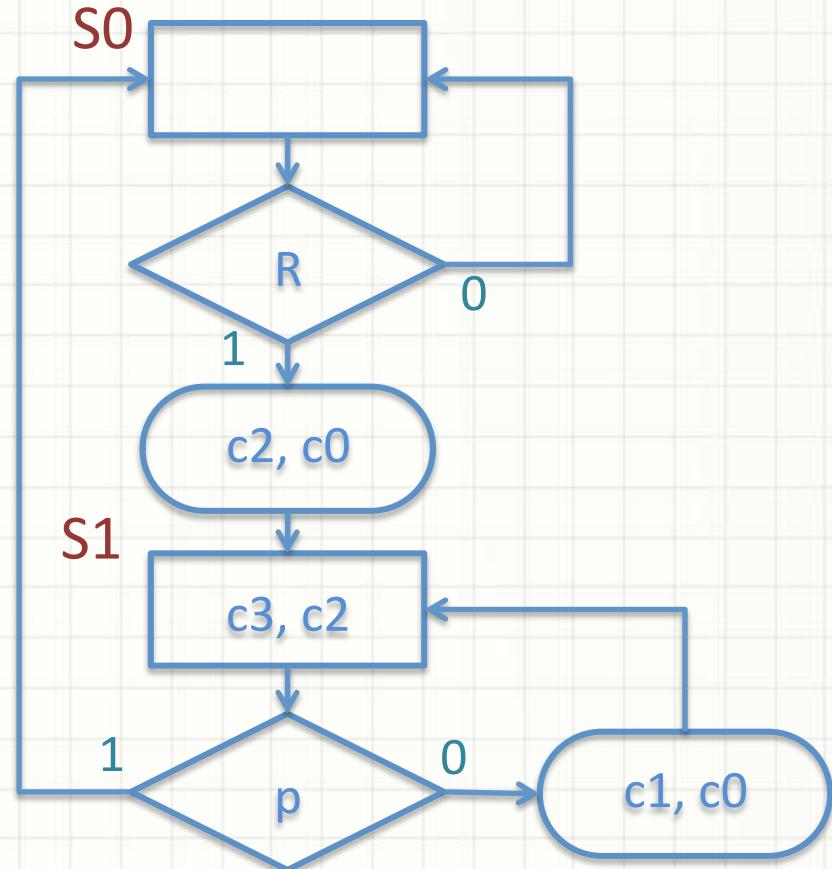
ASM charts with and without data

ASM = Algorithmic State Machine

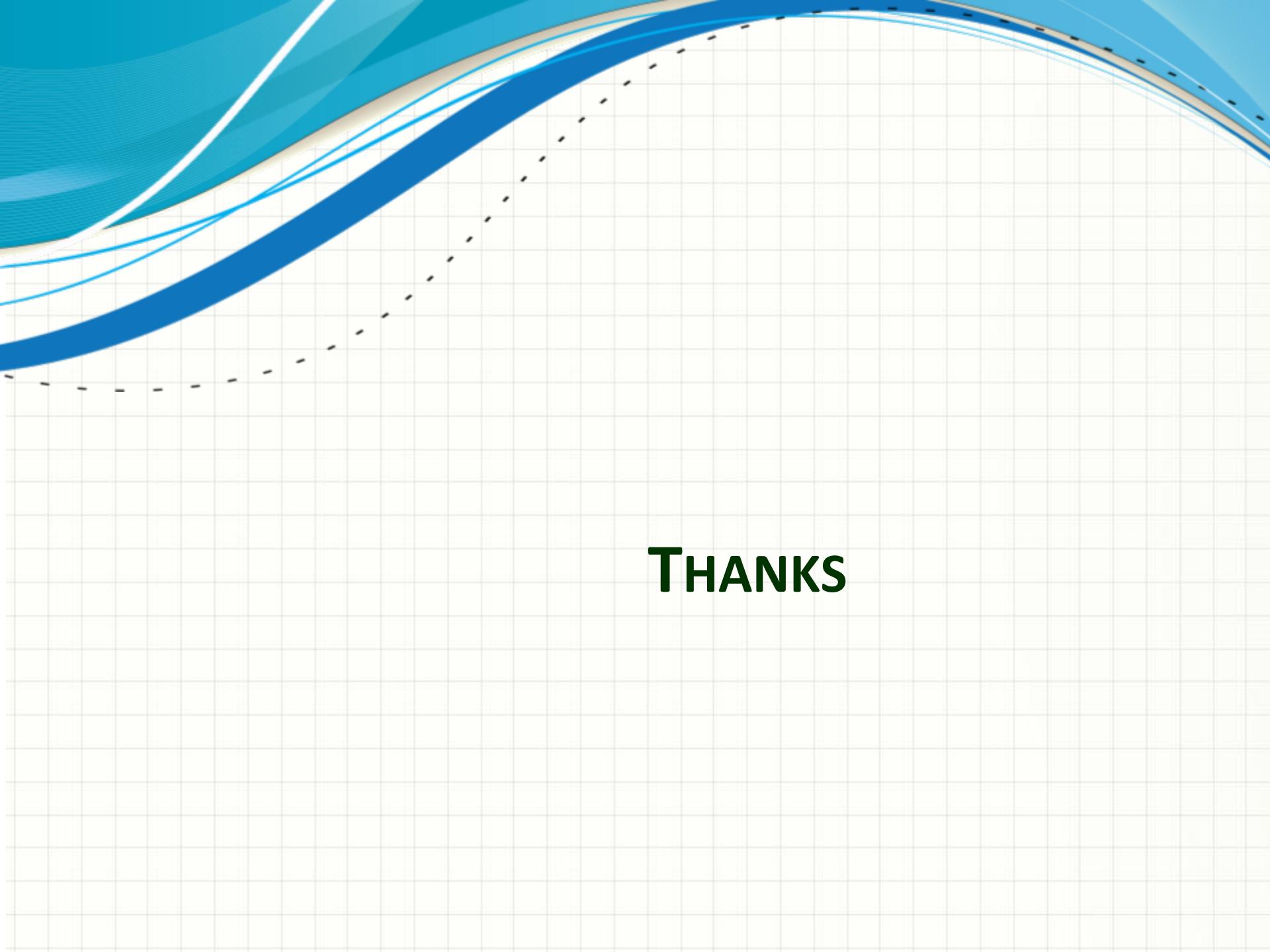


control + data

why control here?



control only

The background features a white grid pattern. Overlaid on the top left is a stylized graphic element consisting of several curved bands in shades of blue and cyan. A prominent thick blue band curves from the bottom left towards the top right. A dashed black line follows a similar path, starting from the bottom left and curving upwards and to the right.

THANKS