Algorithmic State Machine (ASM) Charts

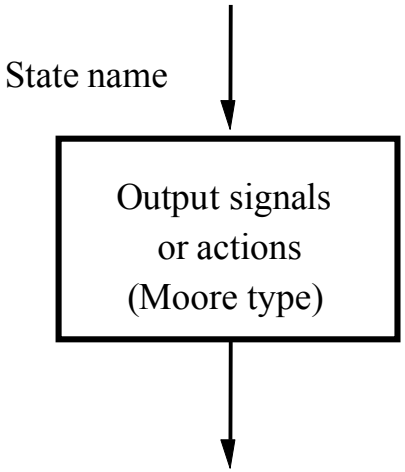
Algorithmic State Machine

Algorithmic State Machine –

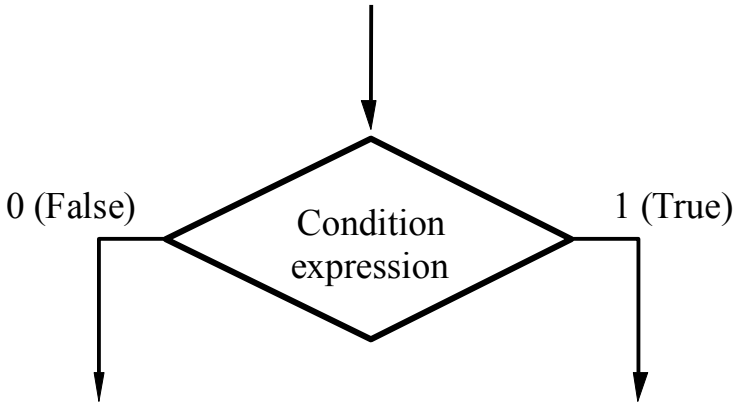
representation of a Finite State Machine

suitable for FSMs with a larger number of inputs and outputs compared to FSMs expressed using state diagrams and state tables.

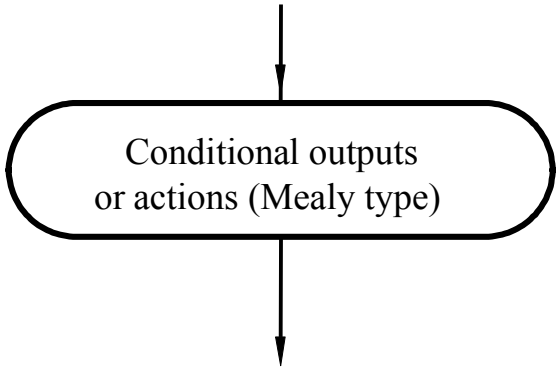
Elements used in ASM charts (1)



(a) State box



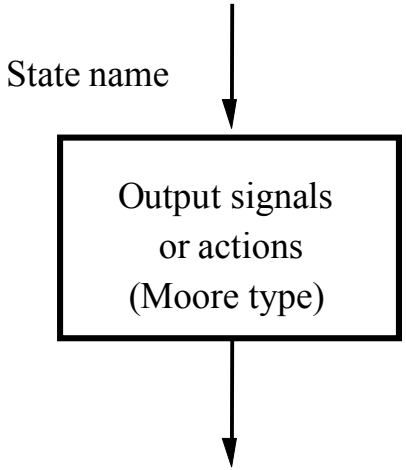
(b) Decision box



(c) Conditional output box

State Box

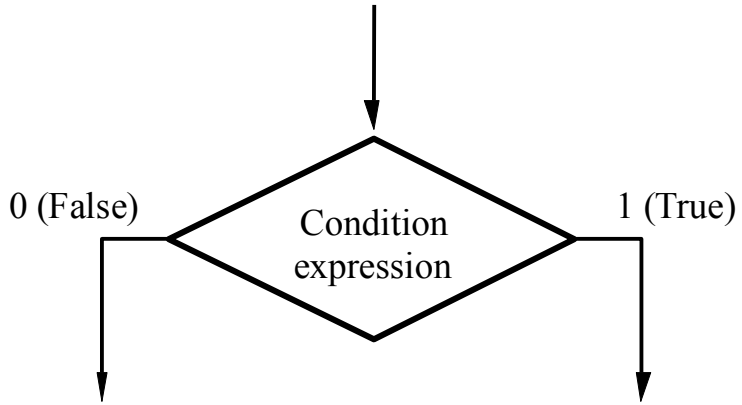
- **State box** – represents a state.
- Equivalent to a node in a state diagram or a row in a state table.
- Contains register transfer actions or output signals
- **Moore-type outputs are listed inside of the box.**
- It is customary to write only the name of the signal that has to be asserted in the given state, e.g., z instead of z<=1.
- Also, it might be useful to write an action to be taken, e.g., count <= count + 1, and only later translate it to asserting a control signal that causes a given action to take place (e.g., enable signal of a counter).



Decision Box

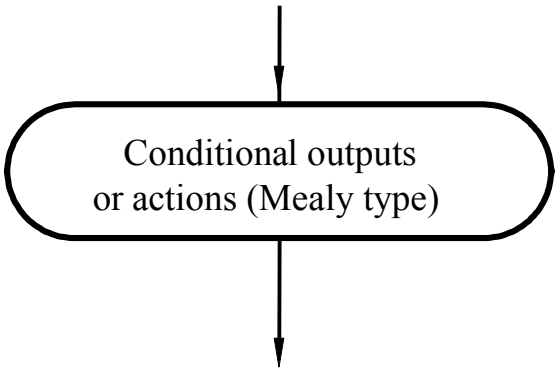
- **Decision box** – indicates that a given condition is to be tested and the exit path is to be chosen accordingly.

  The condition expression may include one or more inputs to the FSM.



Conditional Output Box

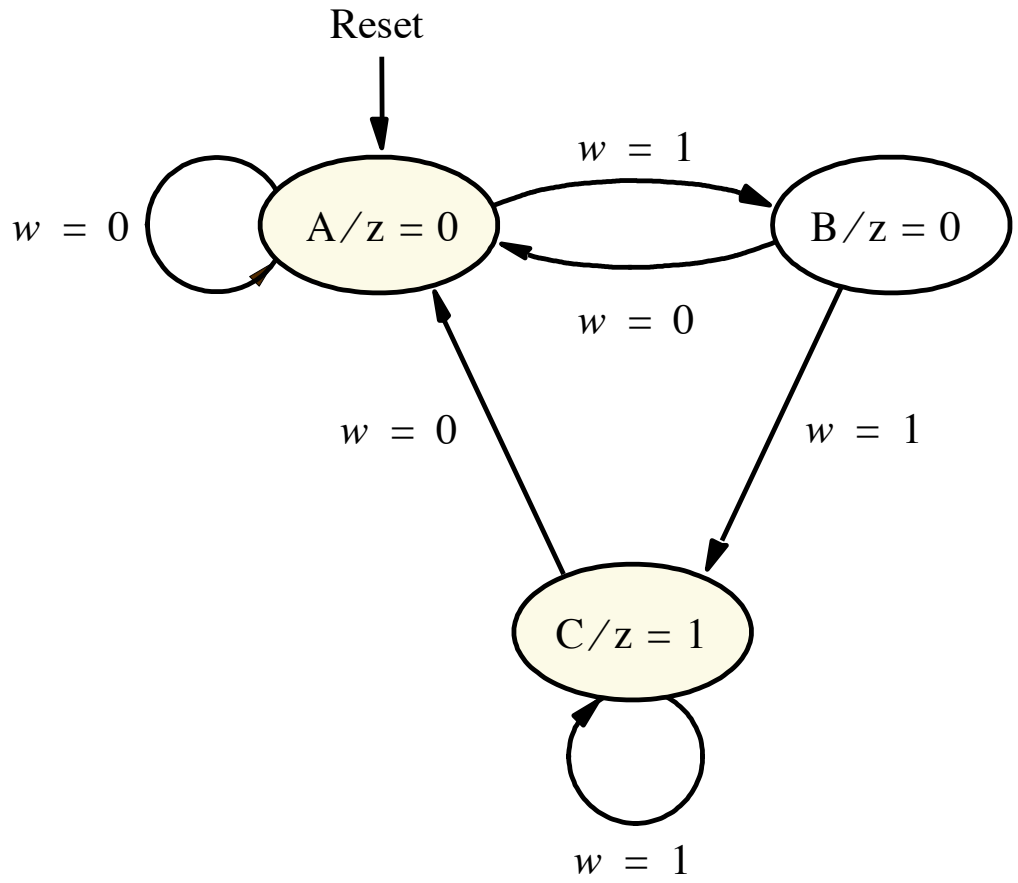
- **Conditional output box**
- Denotes output signals that are of the Mealy type.
- The condition that determines whether such outputs are generated is specified in the decision box.



ASMs representing simple FSMs

- Algorithmic state machines can model both Mealy and Moore Finite State Machines
- They can also model machines that are of the mixed type

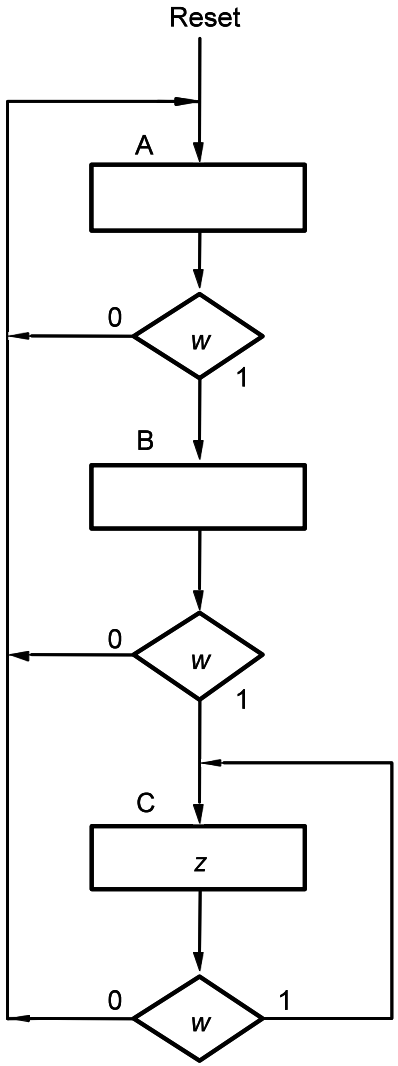
Moore FSM – Example 2: State diagram



Moore FSM – Example 2: State table

Present state	Next state		Output
	$w = 0$	$w = 1$	z
A	A	B	0
B	A	C	0
C	A	C	1

ASM Chart for Moore FSM – Example 2



Example 2: VHDL code (1)

```
USE ieee.std_logic_1164.all ;

ENTITY simple IS
      PORT (  clock   : IN STD_LOGIC ;
              resetn  : IN STD_LOGIC ;
              w       : IN STD_LOGIC ;
              z       : OUT STD_LOGIC ) ;
END simple ;

ARCHITECTURE Behavior OF simple IS
      TYPE State_type IS (A, B, C) ;
      SIGNAL y : State_type ;
BEGIN
      PROCESS ( resetn, clock )
      BEGIN
            IF resetn = '0' THEN
                  y <= A ;
            ELSIF (Clock'EVENT AND Clock = '1') THEN
```

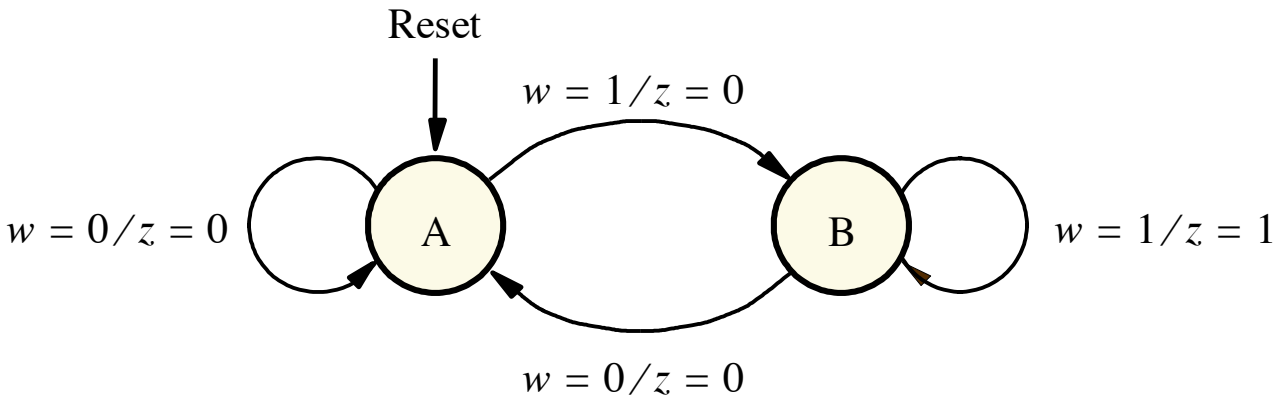
Example 2: VHDL code (2)

```
CASE y IS
    WHEN A =>
        IF w = '0' THEN
            y <= A ;
        ELSE
            y <= B ;
        END IF ;
    WHEN B =>
        IF w = '0' THEN
            y <= A ;
        ELSE
            y <= C ;
        END IF ;
    WHEN C =>
        IF w = '0' THEN
            y <= A ;
        ELSE
            y <= C ;
        END IF ;
END CASE ;
```

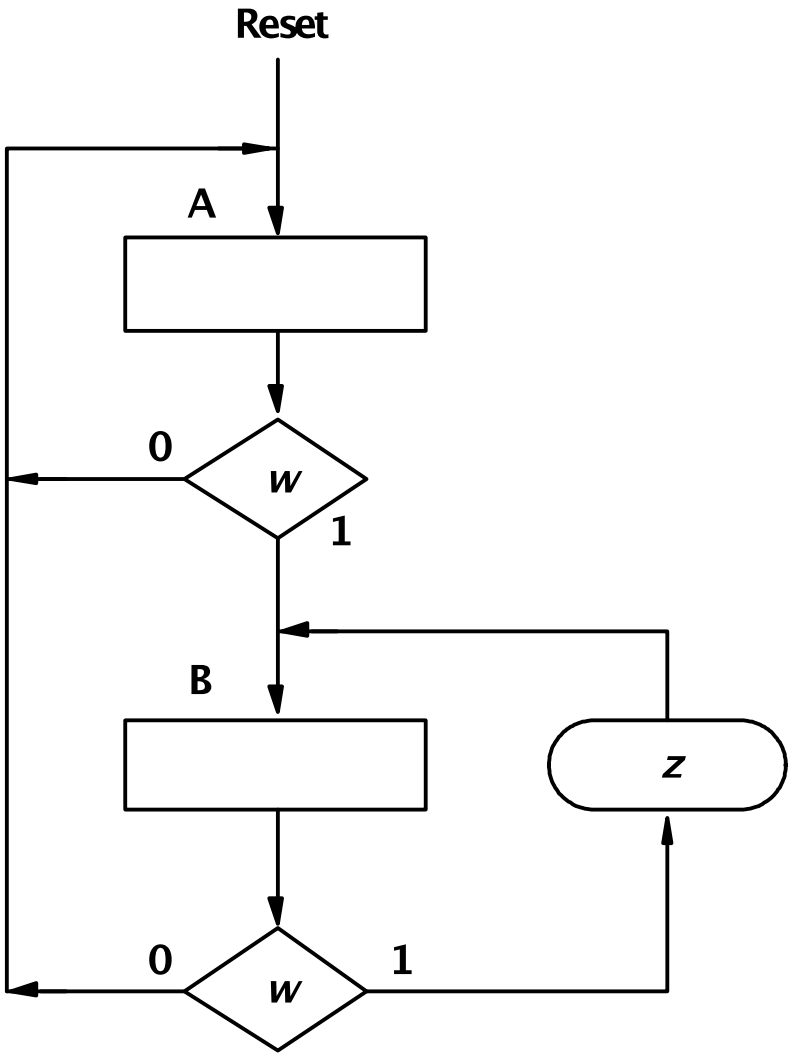
Example 2: VHDL code (3)

```
        END IF ;
    END PROCESS ;

    z <= '1' WHEN y = C ELSE '0' ;

END Behavior ;
```

Mealy FSM – Example 3: State diagram



ASM Chart for Mealy FSM – Example 3



Example 3: VHDL code (1)

```
LIBRARY ieee ;
USE ieee.std_logic_1164.all ;

ENTITY Mealy IS
      PORT ( clock    : IN      STD_LOGIC ;
             resetn   : IN      STD_LOGIC ;
             w        : IN      STD_LOGIC ;
             z        : OUT     STD_LOGIC ) ;
END Mealy ;

ARCHITECTURE Behavior OF Mealy IS
      TYPE State_type IS (A, B) ;
      SIGNAL y : State_type ;
BEGIN
      PROCESS ( resetn, clock )
      BEGIN
          IF resetn = '0' THEN
              y <= A ;
          ELSIF (clock'EVENT AND clock = '1') THEN
```

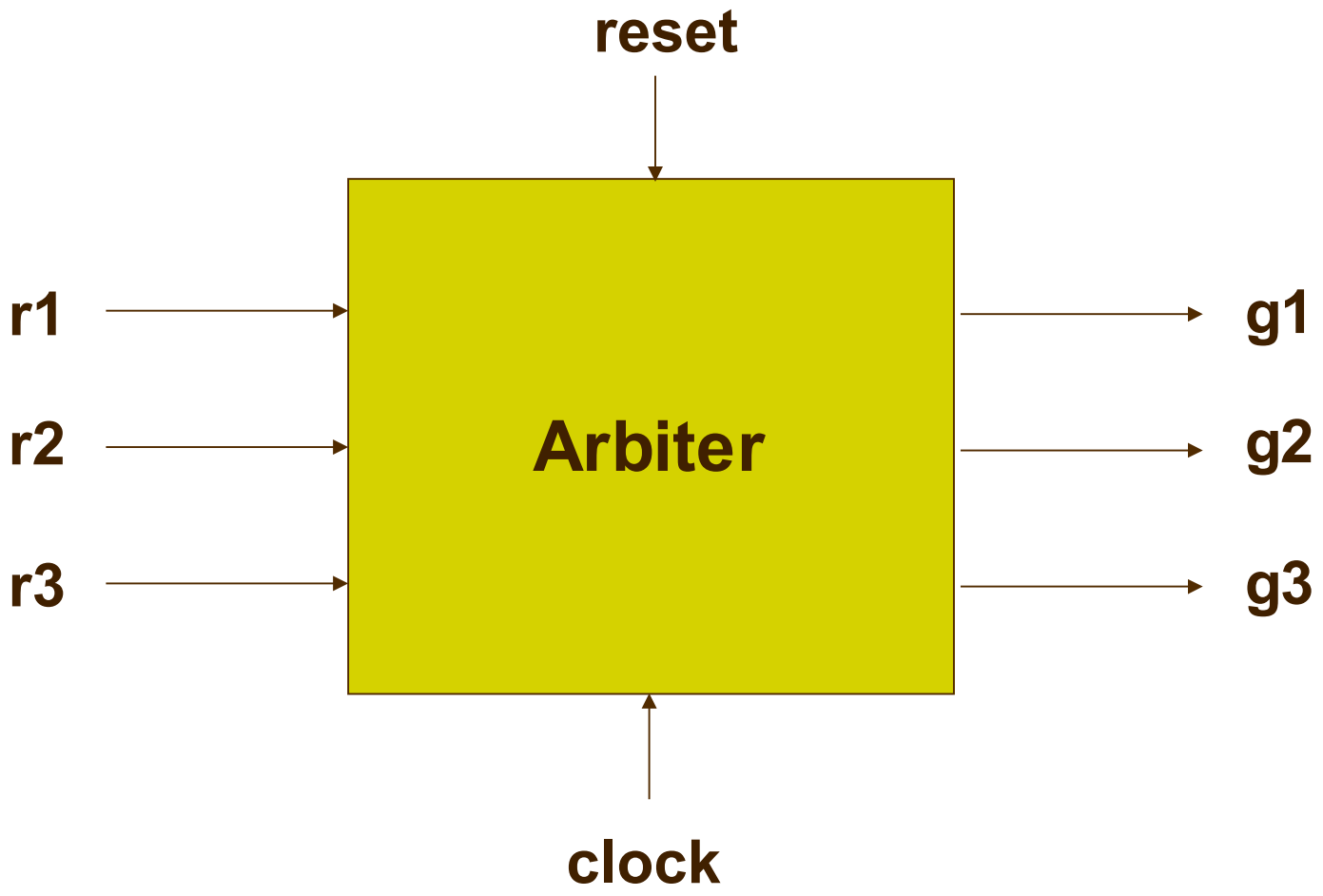
Example 3: VHDL code (2)

```
CASE y IS
    WHEN A =>
        IF w = '0' THEN
            y <= A ;
        ELSE
            y <= B ;
        END IF ;
    WHEN B =>
        IF w = '0' THEN
            y <= A ;
        ELSE
            y <= B ;
        END IF ;
END CASE ;
```

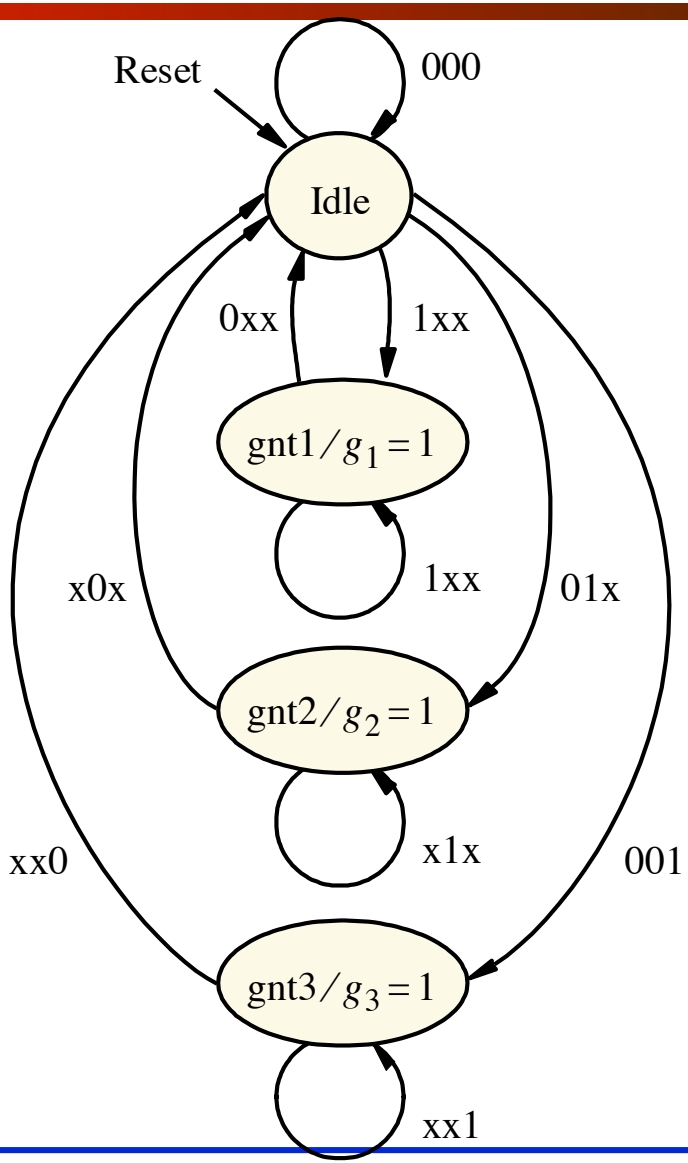
Example 3: VHDL code (3)

```
        END IF ;
    END PROCESS ;

    z <= '1' WHEN (y = B) AND (w='1') ELSE '0' ;

END Behavior ;
```

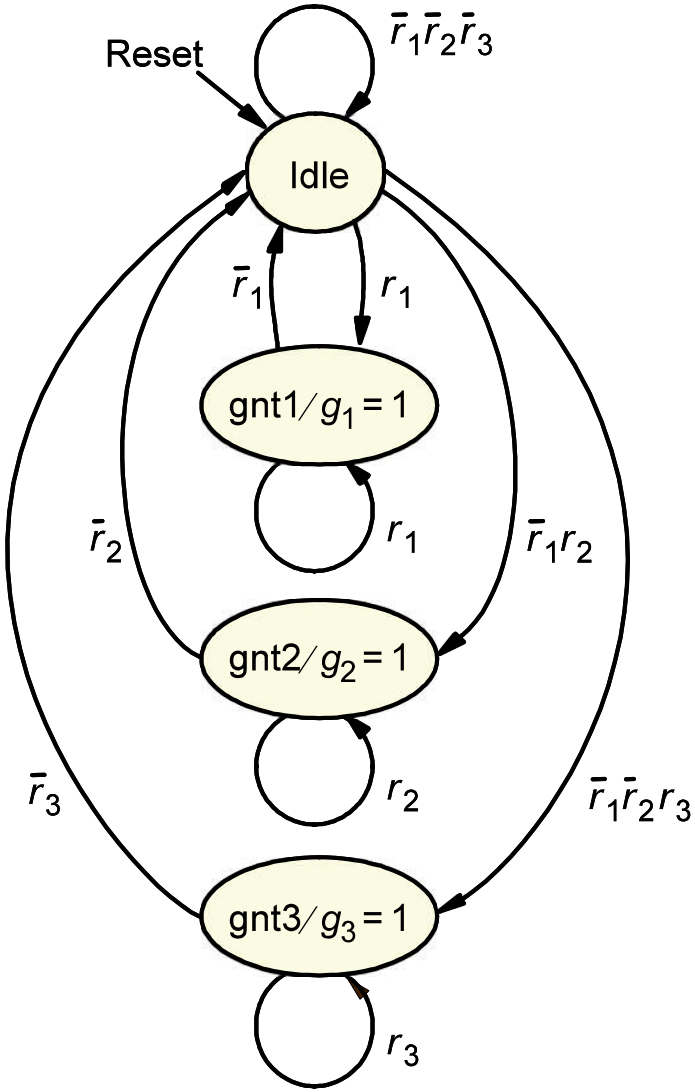
Control Unit Example: Arbiter (1)



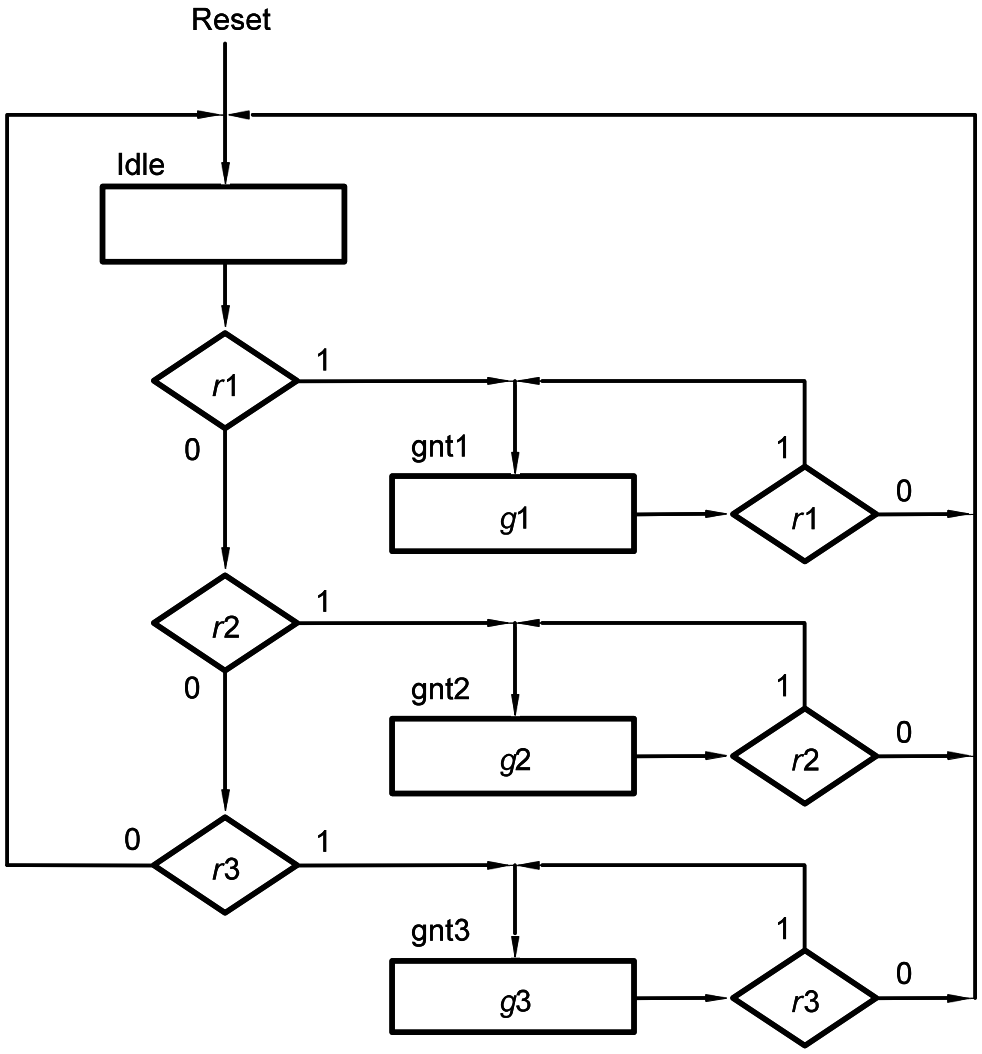
Control Unit Example: Arbiter (2)



Control Unit Example: Arbiter (3)



ASM Chart for Control Unit - Example 4



Example 4: VHDL code (1)

```
LIBRARY ieee;
USE ieee.std_logic_1164.all;

ENTITY arbiter IS
   PORT ( Clock, Resetn    : IN     STD_LOGIC ;
          r                : IN     STD_LOGIC_VECTOR(1 TO 3) ;
          g                : OUT    STD_LOGIC_VECTOR(1 TO 3) ) ;
END arbiter ;

ARCHITECTURE Behavior OF arbiter IS
   TYPE State_type IS (Idle, gnt1, gnt2, gnt3) ;
   SIGNAL y : State_type ;
```

Example 4: VHDL code (2)

```
BEGIN
    PROCESS ( Resetn, Clock )
    BEGIN
            IF Resetn = '0' THEN y <= Idle ;
            ELSIF (Clock'EVENT AND Clock = '1') THEN
                    CASE y IS
                            WHEN Idle =>
                                    IF r(1) = '1' THEN y <= gnt1 ;
                                    ELSIF r(2) = '1' THEN y <= gnt2 ;
                                    ELSIF r(3) = '1' THEN y <= gnt3 ;
                                    ELSE y <= Idle ;
                                    END IF ;
                            WHEN gnt1 =>
                                    IF r(1) = '1' THEN y <= gnt1 ;
                                    ELSE y <= Idle ;
                                    END IF ;
                            WHEN gnt2 =>
                                    IF r(2) = '1' THEN y <= gnt2 ;
                                    ELSE y <= Idle ;
                                    END IF ;
```

Example 4: VHDL code (3)

```
                                    WHEN gnt3 =>
                                            IF r(3) = '1' THEN y <= gnt3 ;
                                            ELSE y <= Idle ;
                                            END IF ;
                            END CASE ;
                    END IF ;
            END PROCESS ;

            g(1) <= '1' WHEN y = gnt1 ELSE '0' ;
            g(2) <= '1' WHEN y = gnt2 ELSE '0' ;
            g(3) <= '1' WHEN y = gnt3 ELSE '0' ;
END Behavior ;
```