

Algorithmic State Machine Charts

Last updated 5/18/20

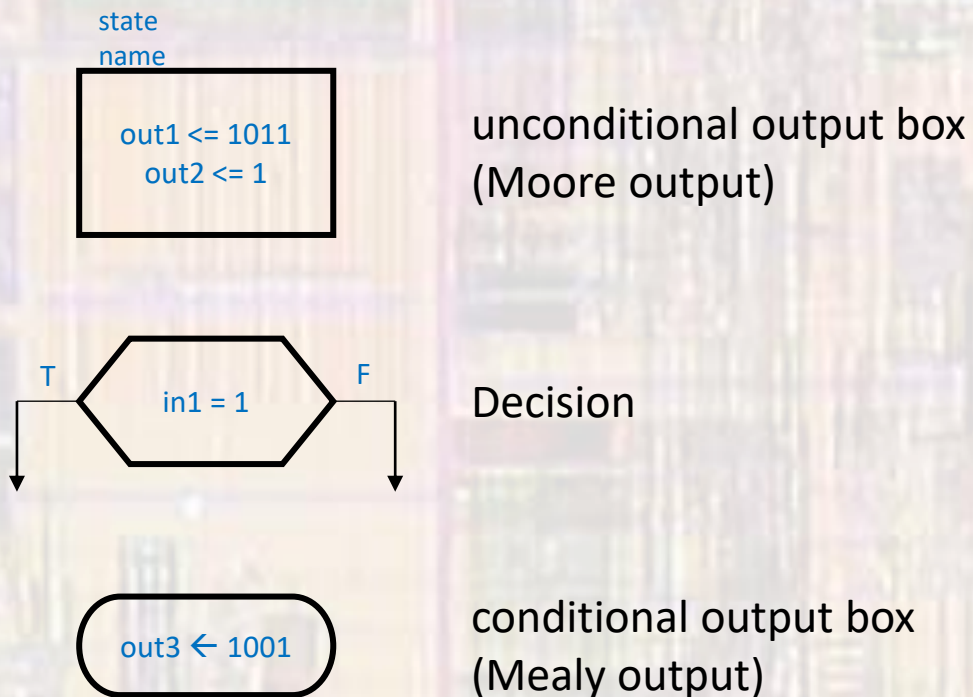
ASM Charts

These slides review the basics of Algorithmic State Machine (ASM) diagrams

Upon completion: You should be able to create your own ASM diagram for a variety of applications

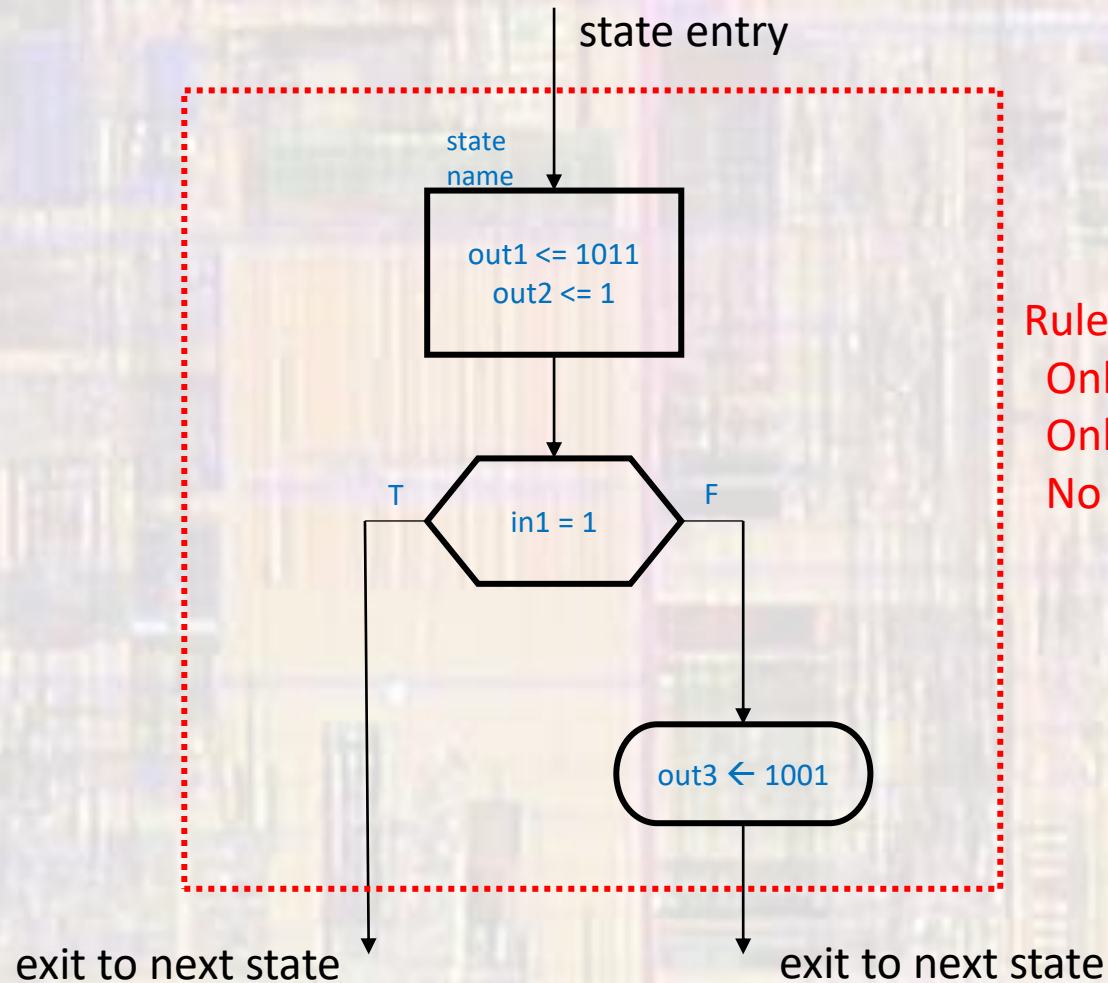
ASM Charts

- Algorithmic State Machine Chart (ASM)
- Alternative representation to a State Diagram



ASM Charts

- Algorithmic State Machine Chart (ASM)

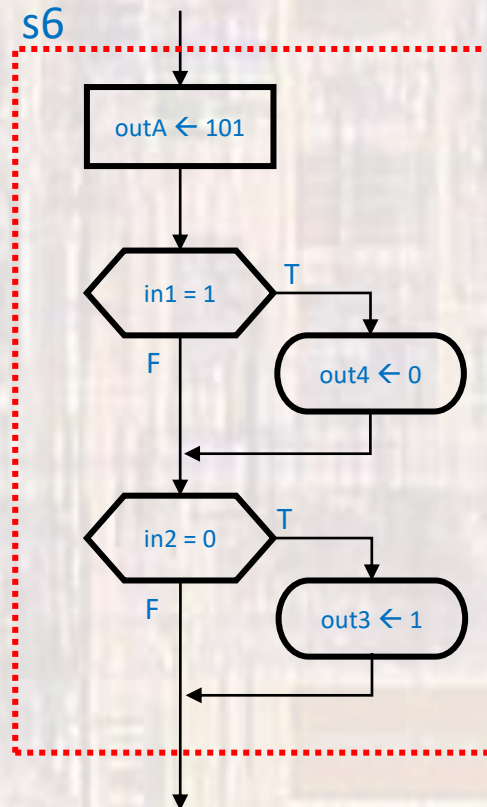
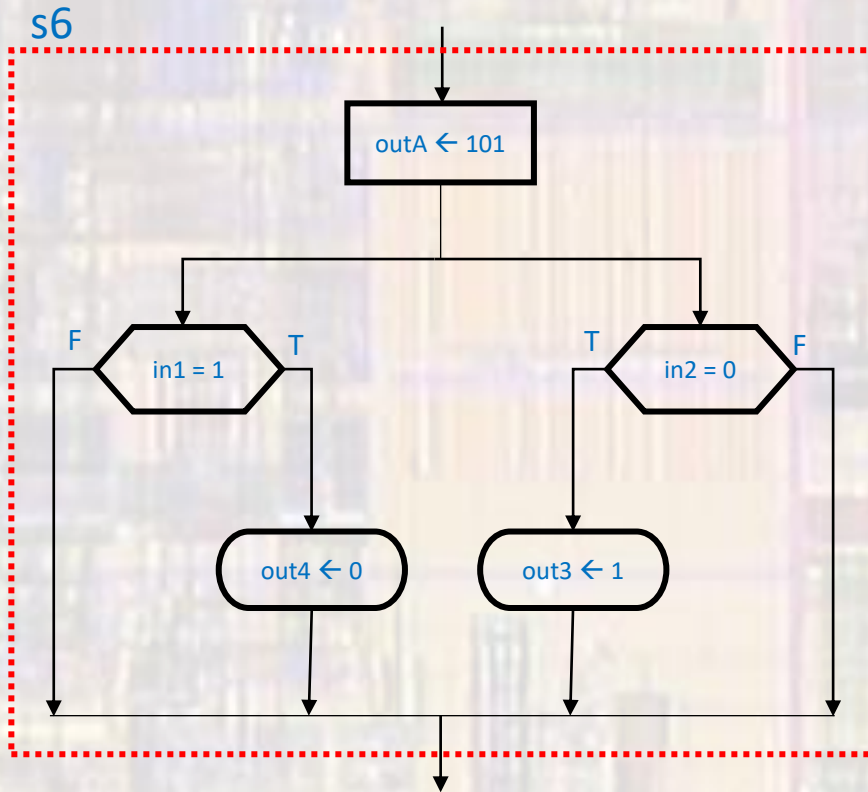


Rules:

- Only one input path
- Only one valid output path
- No internal feedback

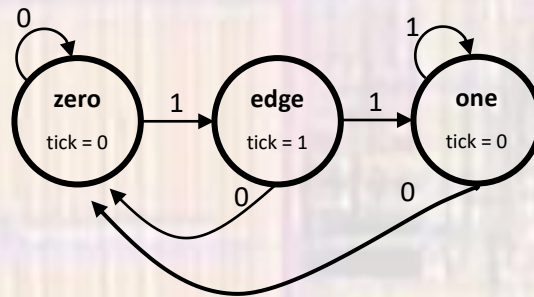
ASM Charts

- Algorithmic State Machine Chart (ASM)



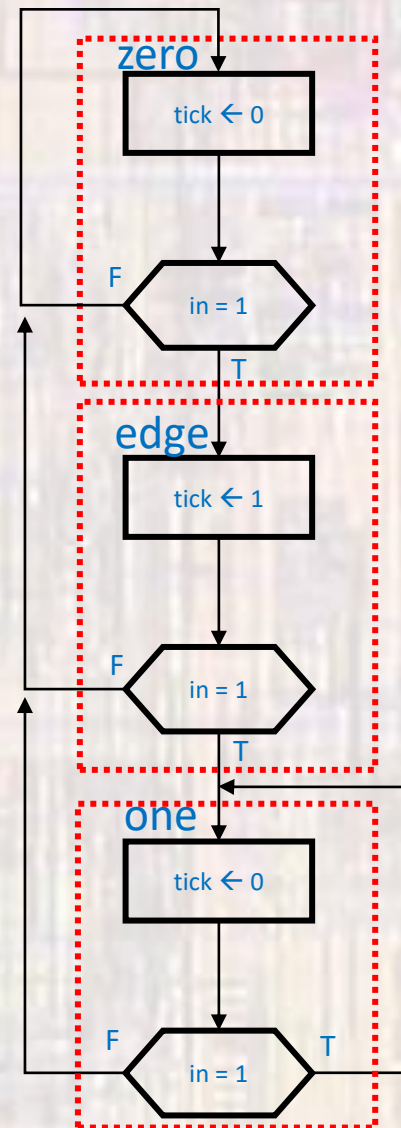
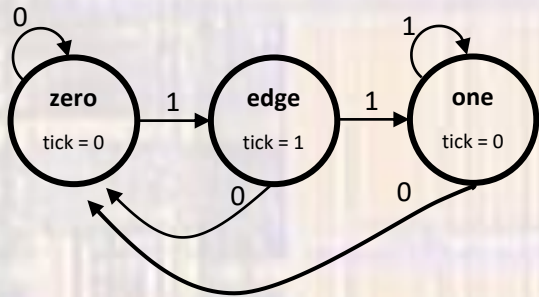
ASM Charts

- Edge Detector
- Rising edge



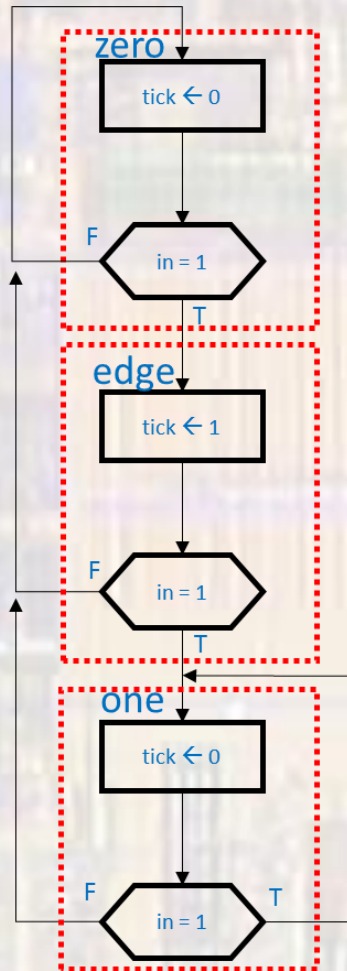
ASM Charts

- Edge Detector



ASM Charts

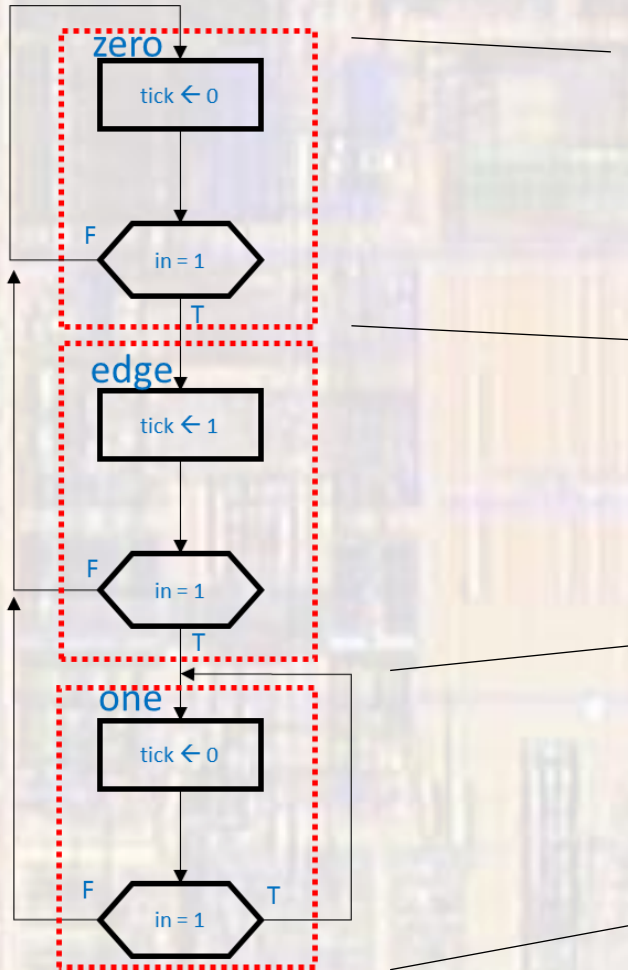
- Edge Detector



```
process(all)
begin
  case state is
    when zero =>
      tick <= '0';
      if in= '1' then
        state_next <= edge;
      else
        state_next <= zero;
      end if;
    when edge =>
      tick <= '1';
      if in= '1' then
        state_next <= one;
      else
        state_next <= zero;
      end if;
    when one =>
      tick <= '0';
      if in= '1' then
        state_next <= one;
      else
        state_next <= zero;
      end if;
  end case;
```


ASM Charts

• Edge Detector



```
-- Next state logic
process(all)
begin
  case state is
    when zero =>
      if in= '1' then
        state_next <= edge;
      else
        state_next <= zero;
      end if;
    when edge =>
      if in= '1' then
        state_next <= one;
      else
        state_next <= zero;
      end if;
    when one =>
      if in= '1' then
        state_next <= one;
      else
        state_next <= zero;
      end if;
  end case;
end case;
```

-- Output logic

```
process(all)
begin
  case state is
    when zero =>
      tick <= '0';
    when edge =>
      tick <= '1';
    when one =>
      tick <= '0';
  end case;
end case;
```