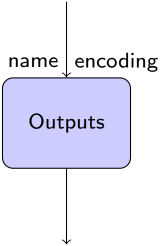
Algorithmic state machines

- An algorithmic state machine (ASM) is simply an alternative to the state diagram — it looks a bit more like a flow chart.
- Consists of three types of boxes.
  - State boxes;
  - Decision boxes;
  - Conditional output boxes.

Algorithmic state machines — state boxes

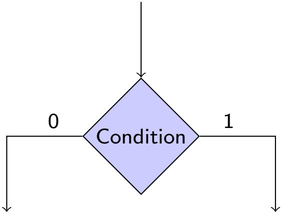
- The state box is rectangular and is equivalent to the state bubble in a state diagram.
- The box has the state name or its binary encoding listed above the box.
- Any outputs that are 1 and that depend only on the state are labeled inside of the box.
- The state box has one entry point and one exit point.
- Example...



- **Note:** Outputs are only shown in the ASM when they are 1.

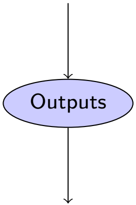
Algorithmic state machines — decision boxes

- The decision box is a diamond shaped box with one entry point and two exist points.
- The inside of the box is labeled with an arbitrary logic expression which evaluates to 0 or 1; the evaluation will cause one the exit points to be chosen.
- Example...



Algorithmic state machines — conditional output boxes

- The conditional output box has one entry and one exit. It specifies an output that occurs when a transition takes place.
- These boxes are required if we have a circuit with Mealy outputs.
- Example...



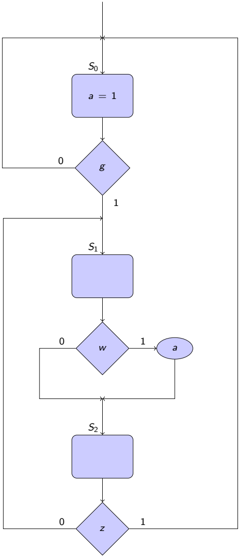
Note: Outputs are only shown in the ASM when they are 1.

Algorithmic state machines

- Designing a circuit given an ASM is exactly the same as if one was given a state diagram.
- You can identify the number of states by the number of state boxes.
- You can determine the next state and transitions based on the decision boxes.
- You can determine the circuit outputs based on the decision boxes and conditional output boxes.
- This will lead to a completed state table and the rest of the design is the same as before (do state assignment, find logic equations, etc.)

Algorithmic state machines

- Example...



Algorithmic state machines

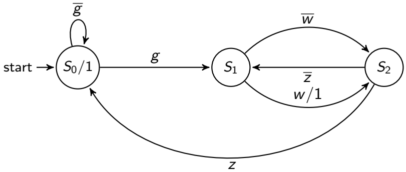
- Example clearly has 3 states S_0, S_1 and S_2 since there are 3 state boxes.
- Example clearly has 3 inputs g, w and z since there are 3 decision boxes and these variables are labeled inside the decision boxes.
- Example clearly has 1 output a which is labeled inside of the state box S_0 and in a conditional output box. Clearly a is 1 when in state S_0 or in state S_1 and $w = 1$, otherwise 0.
- Symbolic state table (written in a slightly different form due to the number of don't care situations in the decision boxes and the large number of inputs):

Current state	**Input**			**Next state**	**Output**
	g	w	z		a
S_0	0	X	X	S_0	1
S_0	1	X	X	S_1	1
S_1	X	0	X	S_2	0
S_1	X	1	X	S_2	1
S_2	X	X	0	S_1	0
S_2	X	X	1	S_0	0

- From here, we perform state assignment, pick a flip flop type, derive next state functions and output functions, etc.

Algorithmic state machines

- You should be able to convert this ASM into a state diagram (and do the reverse procedure if required).



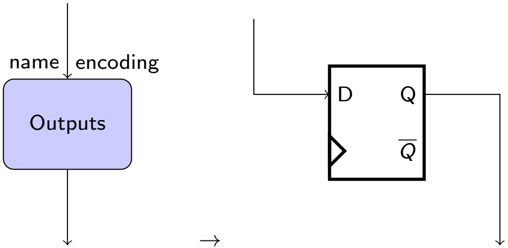
- Note this diagram is sort of a hybrid — sometimes output is shown inside the state bubble and sometimes on the edges.

Algorithmic state machines — one hot encoding

- It is particularly easy to implement an ASM using one hot encoding because every box in the ASM corresponds to a circuit element.
- We might need some additional **OR** gates here and there.

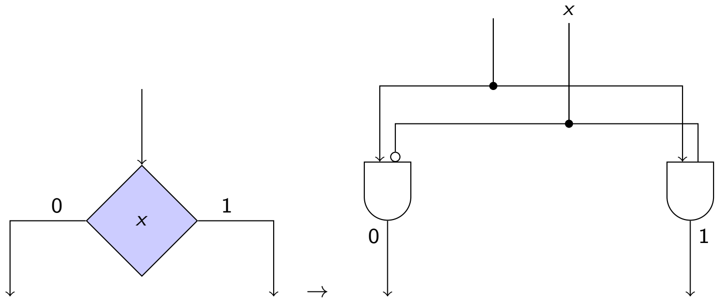
Algorithmic state machines — one hot encoding

- State box...



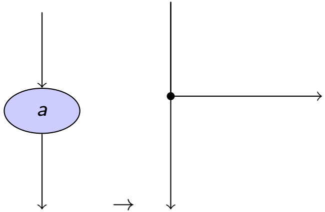
Algorithmic state machines — one hot encoding

- Decision box...



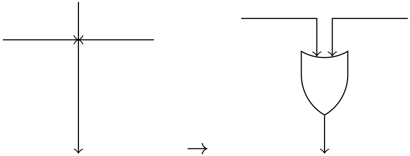
Algorithmic state machines — one hot encoding

- Conditional output box...



We simply "tap off" from the wire passing through the conditional output box.

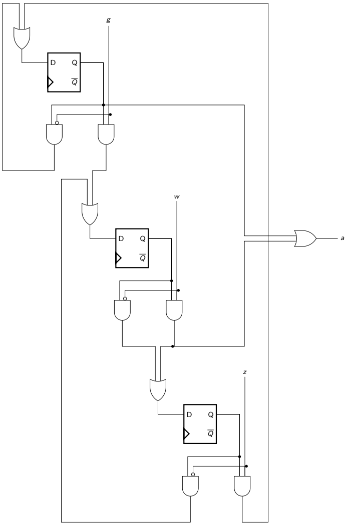
- Joining...



Joining arrows become and **OR** gate.

Algorithmic state machines — one hot encoding

- The circuit we would get from our ASM...



- Of course, you can look at the ASM and write down equations for the *DFF* inputs and circuit outputs too.