Recitation Class 7

VE270 TA Group

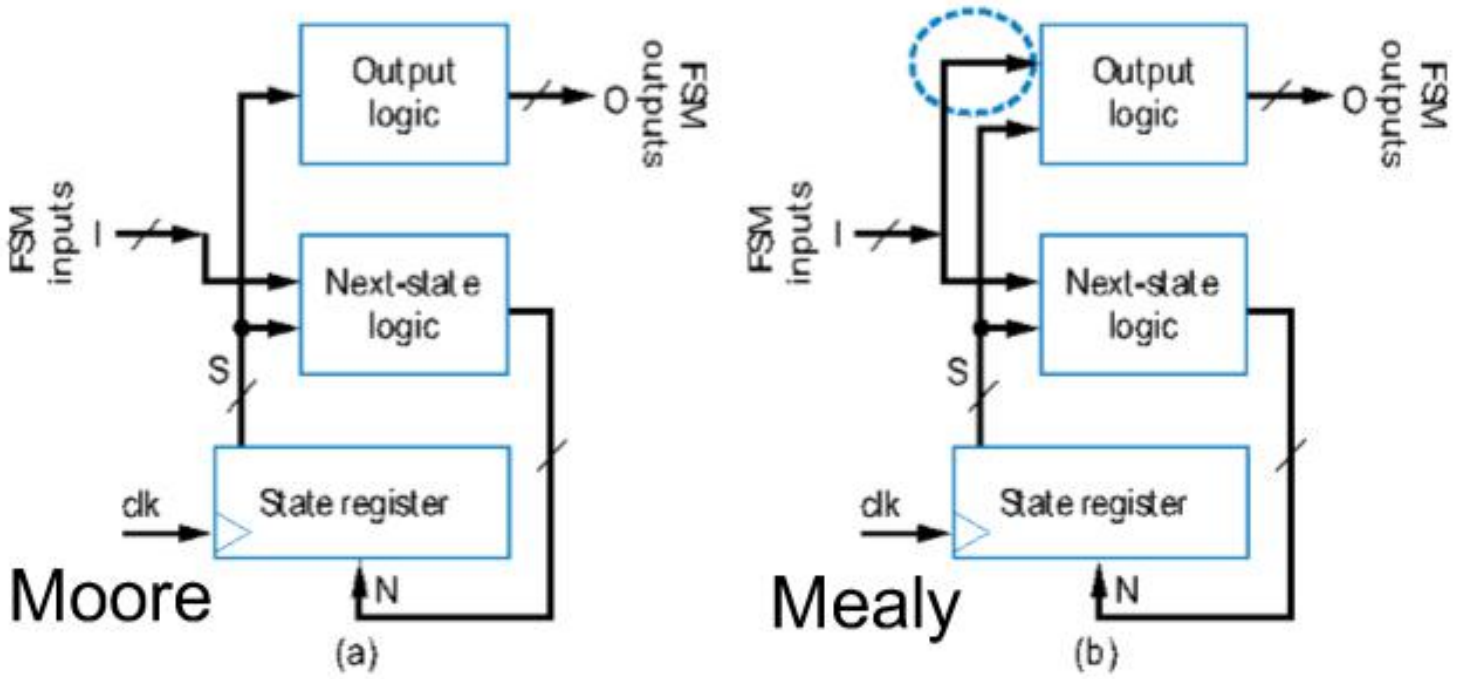
2020.7.15

Outline

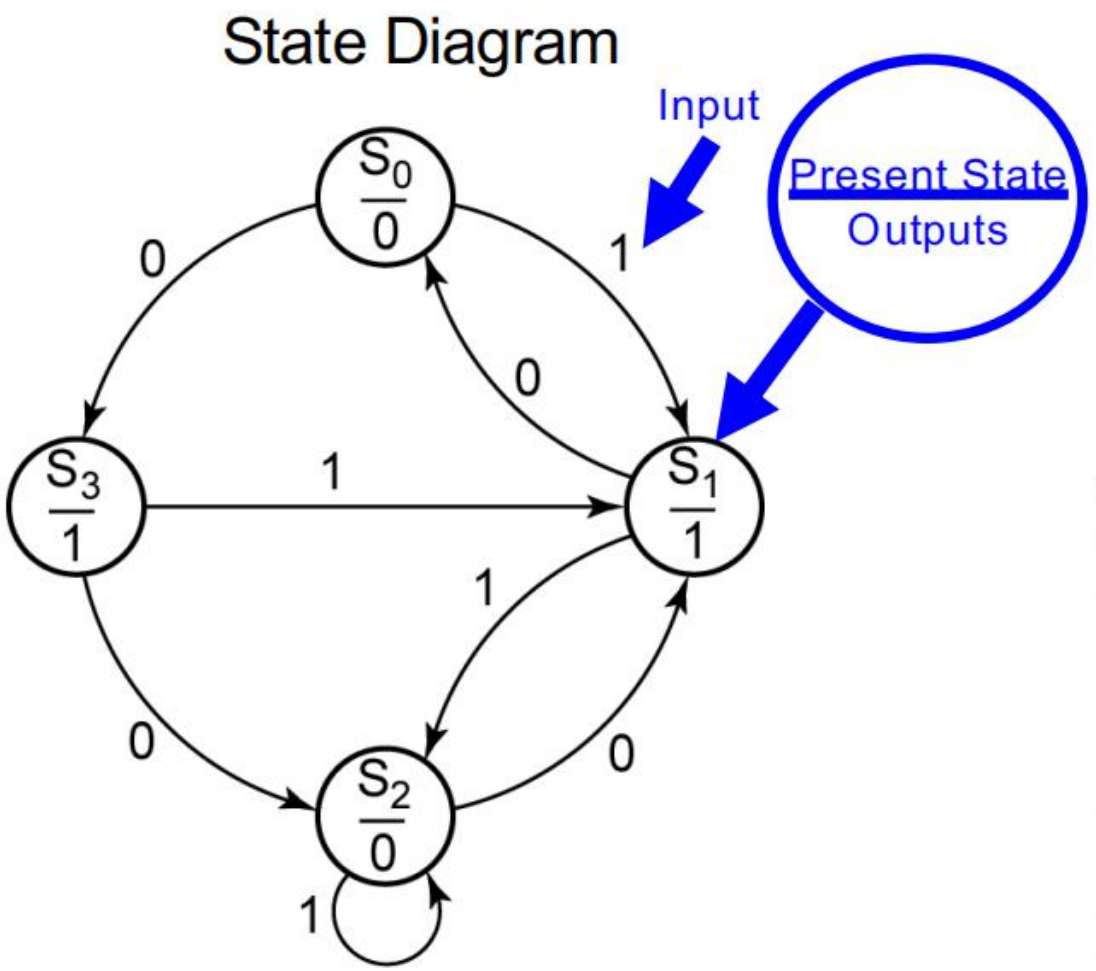
1. Moore and Mealy
2. Implication tables
3. State encoding
4. Self-starting

Moore and Mealy

- Moore: output depends only on the present state
- Mealy: both on the present state and the inputs



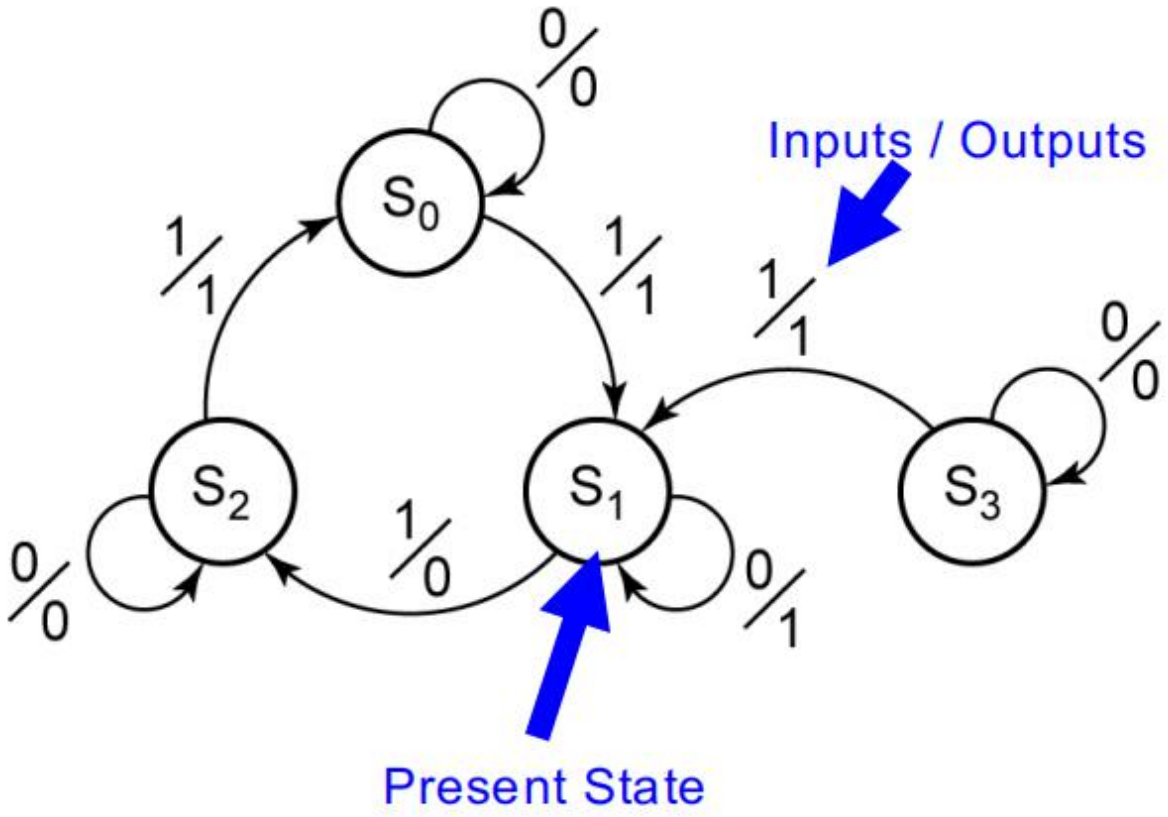
Moore FSM Representation



In	P.S.	N.S.	Out
0	S0	S3	0
1	S0	S1	
0	S1	S0	1
1	S1	S2	
0	S2	S1	0
1	S2	S2	
0	S3	S2	1
1	S3	S1	

Mealy FSM Representation

State Diagram



State Table

In	P.S.	N.S.	Out
0	S0	S0	0
1	S0	S1	1
0	S1	S1	1
1	S1	S2	0
0	S2	S2	0
1	S2	S0	1
0	S3	S3	0
1	S3	S1	1

Comparison

Output

- Mealy: depends on both inputs and presents
- Moore: doesn't depend on inputs

State Diagram

- Mealy: less states -> potentially less number of flip-flops
- Moore: more states than Mealy -> possibly bigger circuit

Speed of output response to the inputs

- Mealy: quick, as soon as input changes
- Moore: as long as one clock cycle delay

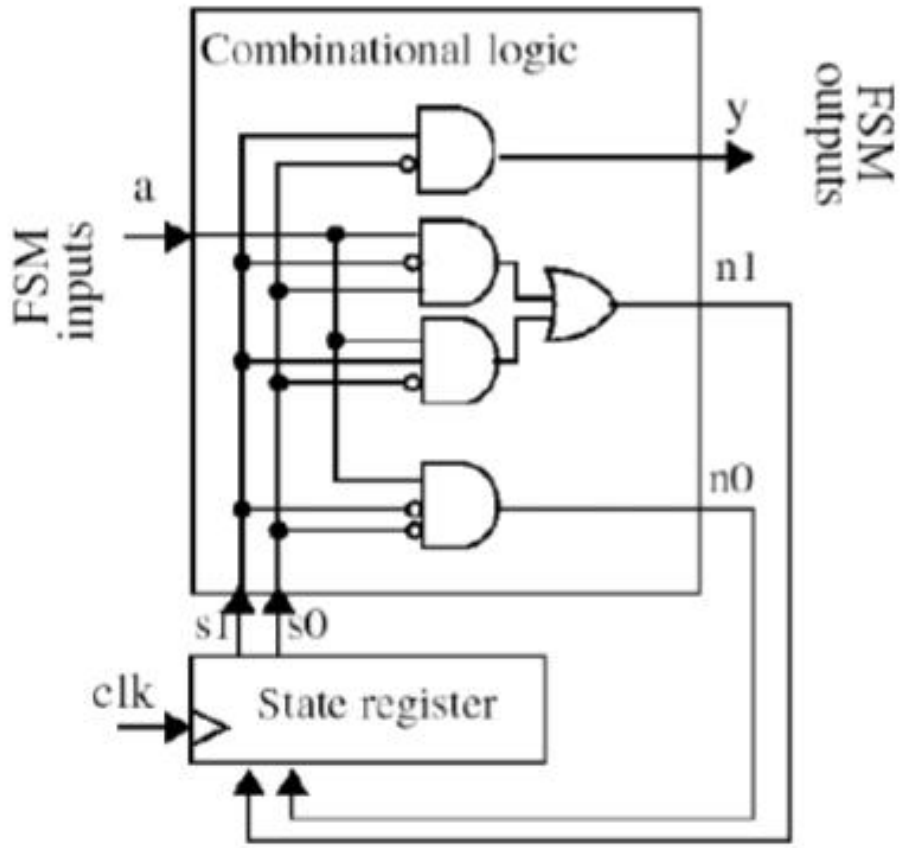
TIMING ISSUE

- Mealy: asynchronous, may cause serious problem
- Moore: synchronous, more stable

Reverse Engineering

Given a circuit of FSM, figure out the behavior

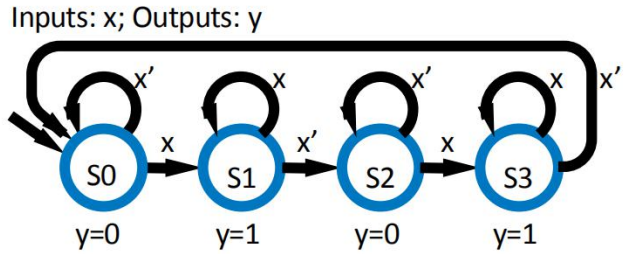
- Mealy or Moore?
- How many states?
- Logic for next state?
- State table?
- State diagram?

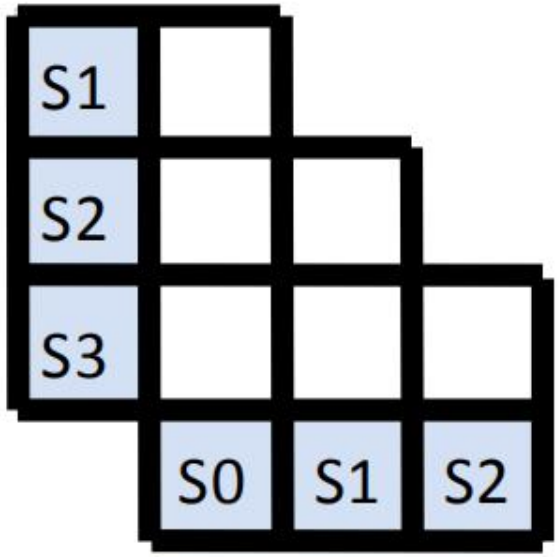


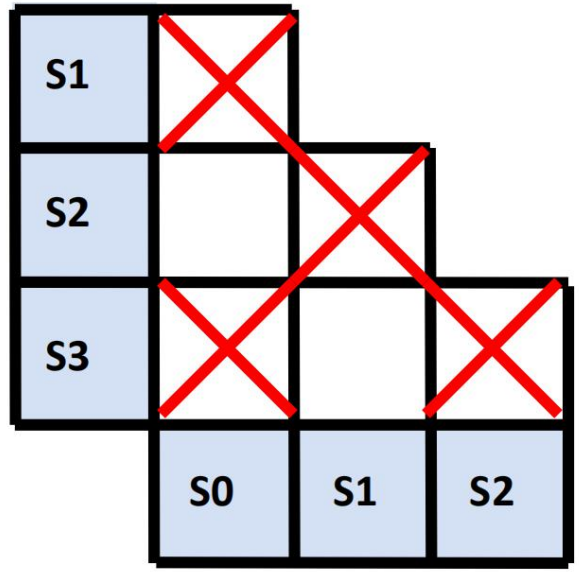
This is the first to do.
Everything will be simple after figuring out the state structure.

Implication table (moore)

1. Construct state pairs
2. Mark out those with different output (for mealy, check all the output under different inputs)





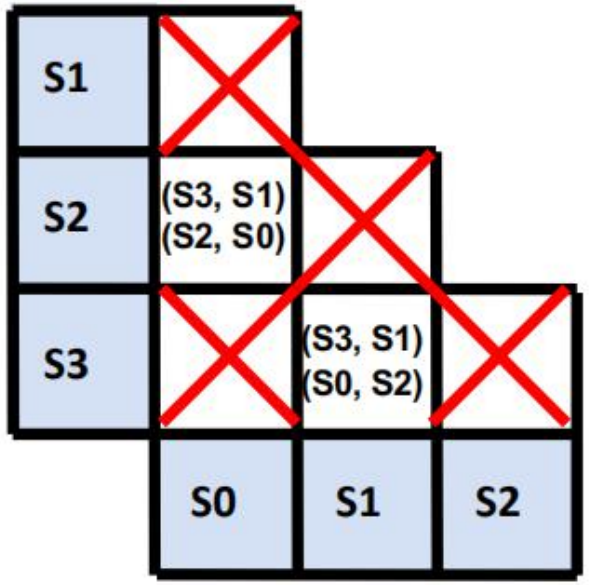


Implication table

3. Check inputs and next state,

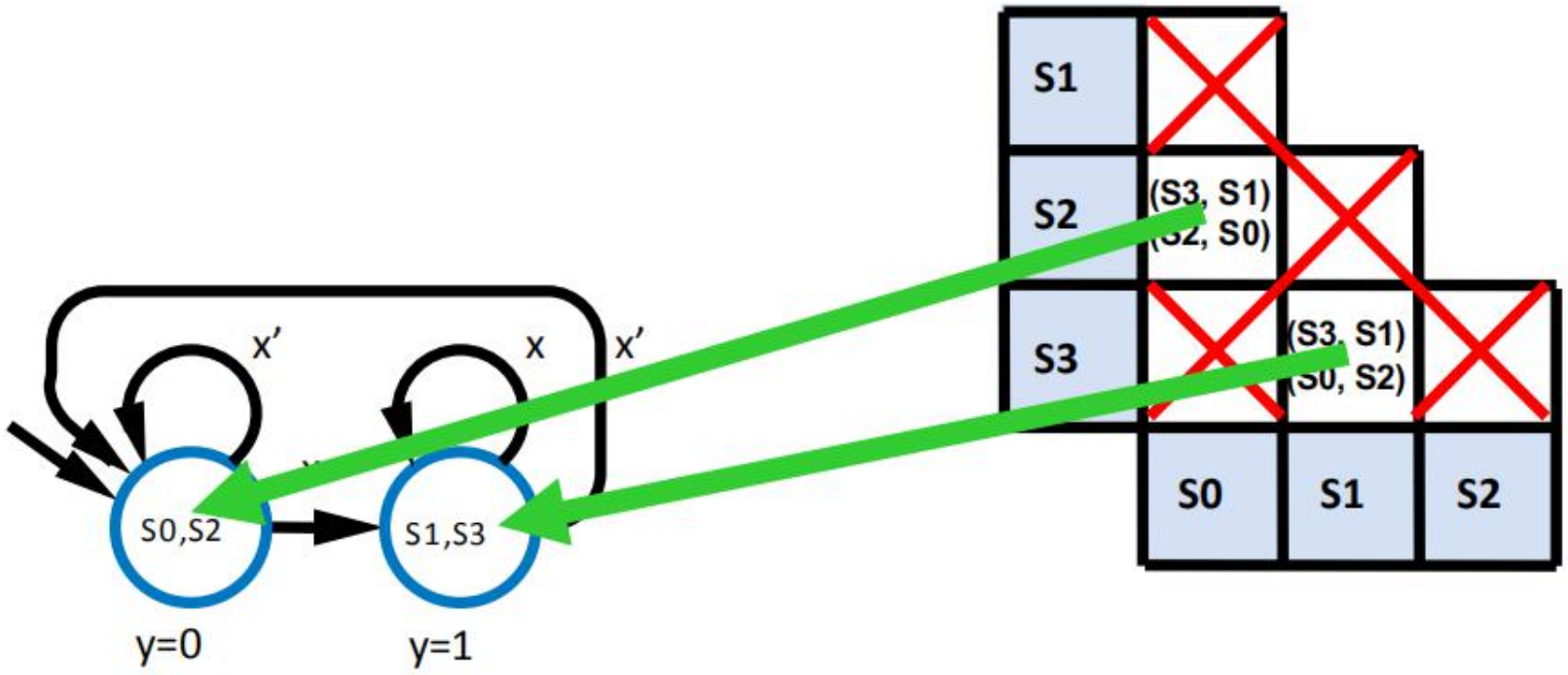
if the pair corresponds to a marked out pair, mark it out.

- **(S2, S0)**
  - Next state pair (**S3**, **S1**) is not marked
  - Next state pair (**S2**, **S0**) is not marked
  - So we do nothing and move on
- **(S3, S1)**
  - Next state pair (**S3**, **S1**) is not marked
  - Next state pair (**S0**, **S2**) is not marked
  - So we do nothing and move on



Implication table

4. Remaining unmarked are equivalent pairs



Example

- Bob is very boring now. He tosses a coin to enjoy himself. Everytime he got a sequence of HHH or HTH (H - head, T - tail), he will be happy and reset the game (to TTT). Suppose the initial sequence is TTT. Design the state diagram and simplify it.

Modeling of FSM

- Mealy: output changes according to input

```
assign out_bit = (((curr_state==zero_2)&&(in_bit==0))||
                  ((curr_state==one_2)&&(in_bit==1))) ? 1 : 0;
```

- Moore: output depends only on current state

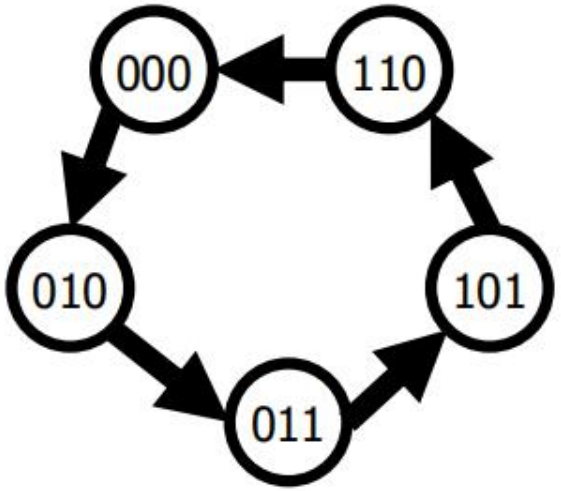
```
assign out_bit = ((curr_state==zero_2)||(curr_state==one_2))
                 ? 1 : 0;
```

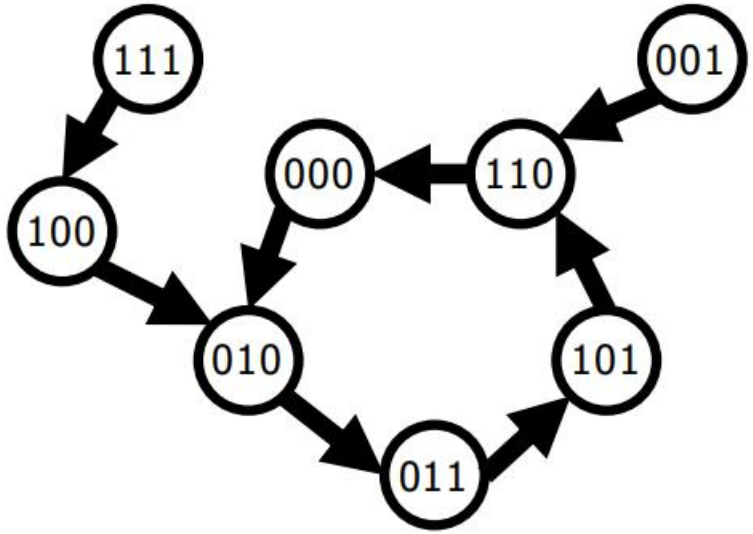
- Notice: Set the output and next state of unused states as 'X'.

State Encoding

- Basic idea: encode the state in the most convenient way
- One hot: 0001, 0010, 0100, 1000, …
- Binary: 0001, 0010, 0011, …

Self-starting





n2 = p0
n1 = p1' + p2'p0'
n0 = p2'p1

1. Set the output and next state of unused states as 'X' to simplify the equation.
2. We need to ensure that the don't care states would be included the loop. (No loop within unused states)

Self-Starting

- Letting the FSM recover from '111' faster.