Centurion
UNIVERSITY
Shaping Lives...
Empowering Communities...

FSM and Efficient Synthesizable FSM Design using Verilog

Centurion
UNIVERSITY
Shaping Lives
Empowering Communities

Introduction

There are many ways to code FSMs including many very poor ways to code FSMs.

This lecture offers guidelines for doing efficient coding, simulation and synthesis of FSM designs.

In this lecture multiple references are made to combinational always blocks and sequential always blocks.

Centurion
UNIVERSITY
Shaping Lives...
Empowering Communities...

Introduction

Combinational *always* blocks are *always* blocks that are used to code combinational logic functionality and are strictly coded using *blocking assignments*. A combinational always block has a combinational sensitivity list, a sensitivity list without *"posedge"* or *"negedge"* Verilog keywords.

Sequential *always* blocks are *always* blocks that are used to code clocked or sequential logic and are always coded using *nonblocking* assignments. A sequential always block has an edge-based sensitivy list.

Centurion
UNIVERSITY
Shaping Lives...
Empowering Communities...

Mealy & Moore FSMs

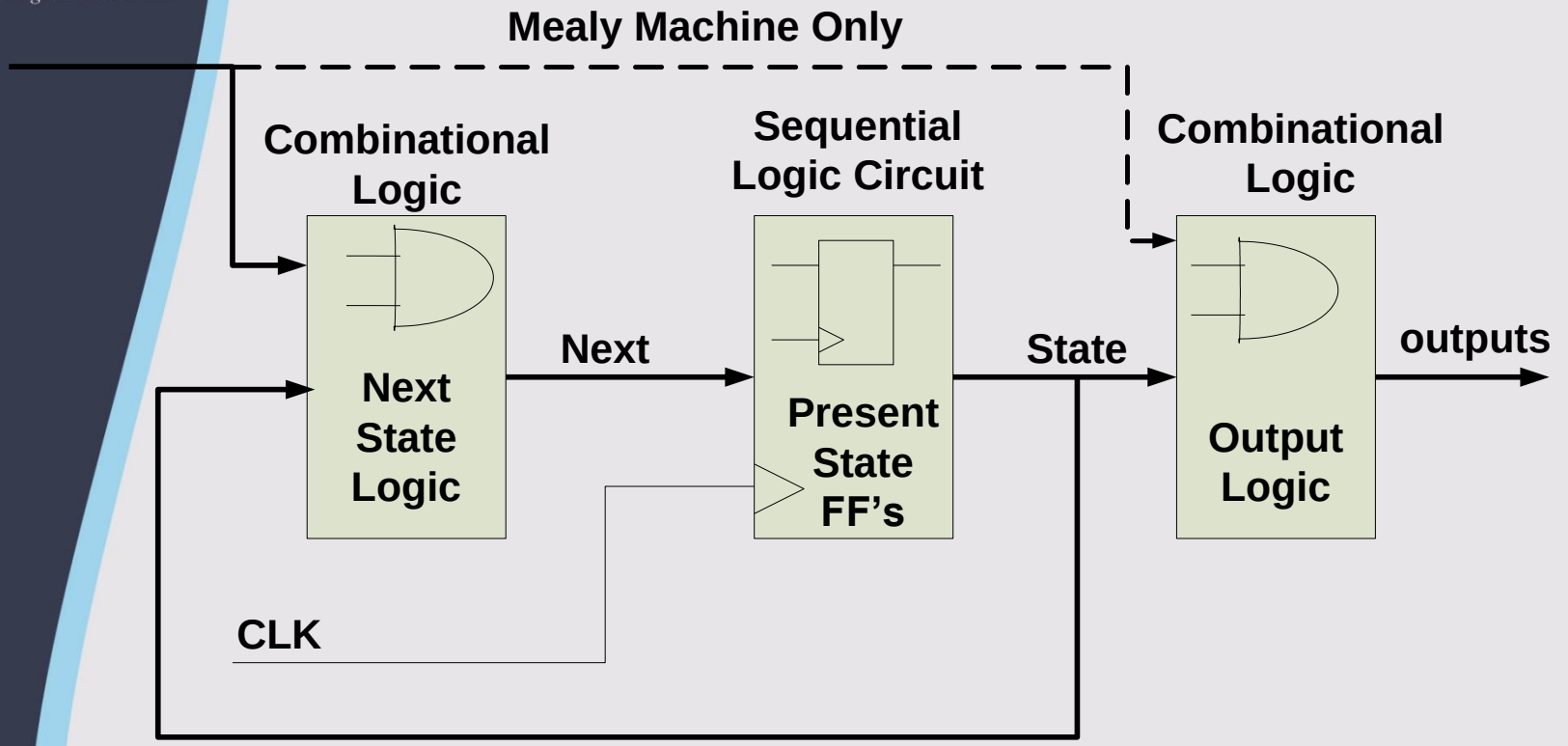
A common classification used to describe the type of an FSM is Mealy and Moore state machines.

A Moore FSM is a state machine where the outputs are only a function of the present state.

A Mealy FSM is a state machine where one or more of the outputs is a function of the present state and one or more of the inputs.

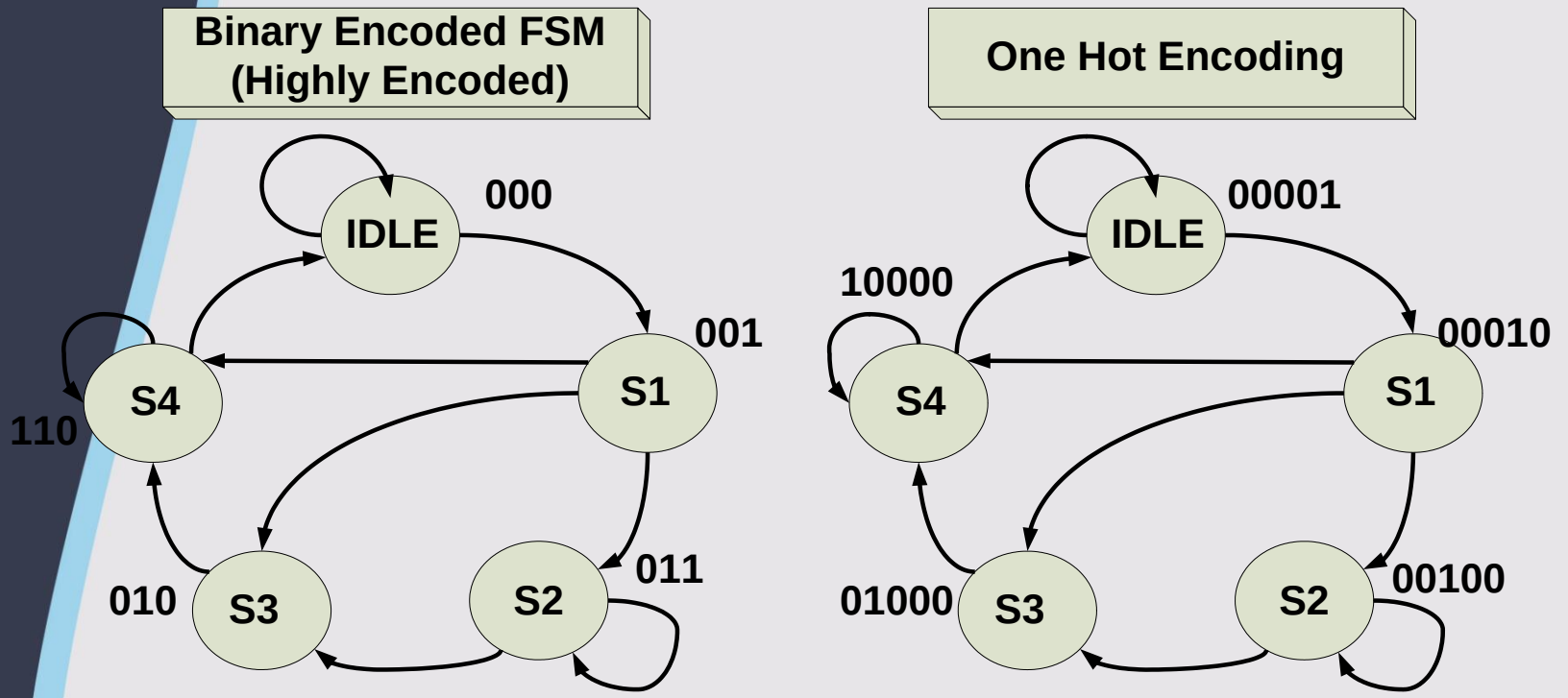
Centurion University
Shaping Lives...
Empowering Communities...

Mealy & Moore FSMs (contd.)



Binary Encoded or One Hot Encoding

Centurion UNIVERSITY
Shaping Lives...
Empowering Communities...



Binary Encoded or One Hot Encoding

A binary-encoded FSM design only requires as many flip-flops as are needed to uniquely encode the number of states in the state machine.

Number of FF

if(log2(number of states) == integer)

required FF = log2(number of states)

else

required FF = integer(log2(#states))+1;

Centurion
UNIVERSITY
Shaping Lives
Empowering Communities

Binary Encoded or One Hot Encoding

A onehot FSM design requires a flip-flop for each state in the design and only one flip-flop (the flip-flop representing the current or "hot" state) is set at a time in a onehot FSM design.

For a state machine with 9-16 states, a binary FSM only requires 4 flip-flops while a onehot FSM requires a flip-flop for each state in the design (9-16 flip-flops).

Centurion
UNIVERSITY
Shaping Lives...
Empowering Communities...

FSM Coding Goals

What constitutes an efficient FSM coding style?

Identify HDL coding goals and why they are important.

Quantify the capabilities of various FSM coding styles.

Centurion
UNIVERSITY
Shaping Lives...
Empowering Communities...

FSM Coding Goals

- The FSM coding style should be easily modified to change state encodings and FSM styles.
- The coding style should be compact.
- The coding style should be easy to code and understand.
- The coding style should facilitate debugging.
- The coding style should yield efficient synthesis results.