

State Machine Encoding HDL

Last updated 1/22/21

State Machine Encoding - HDL

- Encoding
 - Comparison – considerations
 - Memory elements (bits) \rightarrow area or fit
 - Coding / Decoding logic \rightarrow area or fit
 - Speed
 - Shorter logic chains \rightarrow faster maximum clock speeds
 - Power
 - $P = v * i$
 - $P = v * c * dv/dt$
 - dv/dt represents a transition $0 \rightarrow 1$
 - Fewer transitions \rightarrow lower power

State Machine Encoding - HDL

- Encoding – Quartus options

Auto	Allows the Compiler to choose the best encoding for the state machine.
Gray	Uses the minimal number of bits to encode the state machine, ceiling of 2 to the log of n states. This setting is difference from the Minimal Bits setting because each state has only one bit difference from its neighboring states.
Johnson	The number of bits used to encode the state machine is the ceiling of half of the states. Each state has only one bit difference from its neighboring states. Each state is generated by shifting the previous state's bits to the right by 1. The MSB of each state is the negation of the LSB of the previous state.
Minimal Bits	Uses the minimal number of bits to encode the state machine.
One-Hot	Encodes the state machine in the one-hot style. For one-hot encoding, the Quartus II software does not guarantee that each state has one bit set to one and all other bits to zero. Instead, the Quartus II software makes sure that the reset state is all-zeroes. All other states have two bits set to one and all others to zero. This is done so the state machine powers up in the reset state. This encoding has the same properties as true one-hot encoding: each state can be recognized by the value of one bit.
Sequential	Uses the minimal number of bits to encode the state machine; each state is the binary form of its state value.
User-Encoded	Encodes the state machine in the manner specified by the user.

State Machine Encoding - HDL

The image shows the Xilinx IDE interface. On the left, the 'Assignments' pane is visible, with a red arrow pointing to the 'class_proj' entry. A red oval highlights the text 'Assignments → settings'. In the center, the 'Settings - class_proj' dialog is open, with 'Compiler Settings' selected in the left-hand menu. A red oval highlights 'Compiler Settings' and another red oval highlights 'Advanced Settings (Synthesis)...'. On the right, the 'Advanced Analysis & Synthesis Settings' dialog is open, showing a table of settings. A red oval highlights the 'State Machine Processing' row, where the 'Setting' dropdown menu is open, showing options: One-Hot, Auto, Gray, Johnson, Minimal Bits, One-Hot, Sequential, and User-Encoded. The 'User-Encoded' option is selected. Below the table, a description box explains the state machine processing style.

Assignments → settings

Name:	Setting:
Pre-Mapping Resynthesis Optimization	Off
Remove Duplicate Registers	On
Remove Redundant Logic Cells	Off
Report Connectivity Checks	On
Report Parameter Settings	On
Report Source Assignments	On
Resource Aware Inference For Block RAM	On
Restructure Multiplexers	Auto
SDC constraint protection	Off
Safe State Machine	Off
Shift Register Replacement - Allow Asynchronous Clear Signal	On
State Machine Processing	One-Hot
Structural RAM Replacement	Auto
Synchronization Register Chain Length	Gray
Synthesis Effort	Johnson
Timing-Driven Synthesis	Minimal Bits
Use LogicLock Constraints during Resource Balancing	One-Hot

Description:
Specifies the processing style used to compile a state machine. You can use your own 'User-Encoded' style, or select 'One-Hot', 'Minimal Bits', 'Gray', 'Johnson', 'Sequential' or 'Auto' (Compiler-selected) encoding.

State Machine Encoding - HDL

Assignments → settings

Compiler Settings

Specify high-level optimization settings control the optimization focus and algorithm.

Optimization mode

- Balanced (Normal flow)
- Performance (High effort - increase compilation time)
- Performance (Aggressive - increase compilation time)
- Power (High effort - increases runtime)
- Power (Aggressive - increases runtime)
- Area (Aggressive - reduces performance)

Prevent register optimizations

- Prevent register retiming

Advanced Settings (Synthesis)...

Description: Controls the Compiler's high-level optimization mode. In balanced mode, targeting the design's total compilation time. High effort and aggressive modes target other optimization metrics (performance, area, power).

Advanced Analysis & Synthesis Settings

Specify the settings for the logic options in your project. Assignments made to an individual node or entity in the Assignment Editor will override the option settings in this dialog box.

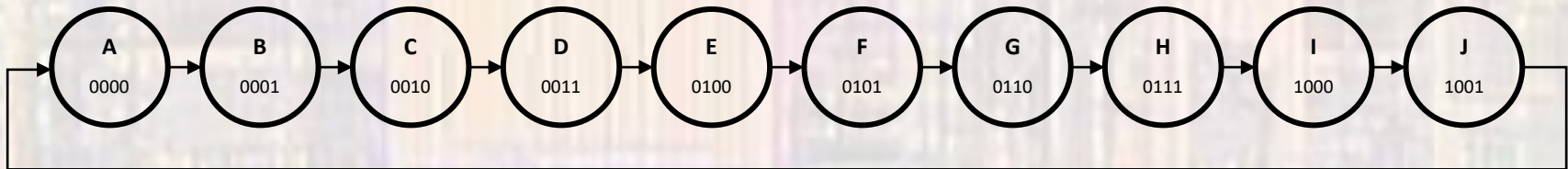
Name:	Setting:
Allow Any RAM Size For Recognition	Off
Allow Any ROM Size For Recognition	Off
Allow Any Shift Register Size For Recognition	Off
Allow Register Duplication	On
Allow Register Merging	On
Allow Shift Register Merging across Hierarchies	Auto
Allow Synchronous Control Signals	On
Analysis & Synthesis Message Level	Medium
Auto Carry Chains	On
Auto Clock Enable Replacement	On
Auto DSP Block Replacement	On
Auto Gated Clock Conversion	Off
Auto Open-Drain Pins	On
Auto RAM Replacement	On
Auto ROM Replacement	On
Auto Resource Sharing	Off
Auto Shift Register Replacement	Auto

Description: Controls whether the Compiler is allowed to duplicate registers to improve design performance. When register duplication is allowed, the Compiler may perform optimizations that create a second copy of a register and move a portion of its fan-out to this new node, in order to improve routability and/or reduce the total routing wire required to route a net with many fan-outs.

Reset, Reset All, OK, Cancel, Help

State Machine Encoding - HDL

- Encoding
 - Comparison – Mod 10 counter



State Machine Encoding - HDL

- Encoding
 - Comparison – Mod 10 counter
10 values (states) to encode
 - Binary (sequential)
0000, 0001, 0010, 0011, 0100, 0101, 0110, 0111, 1000, 1001
 - 4 registers
 - complex decoding
 - slower
 - bigger
 - multiple bit transitions

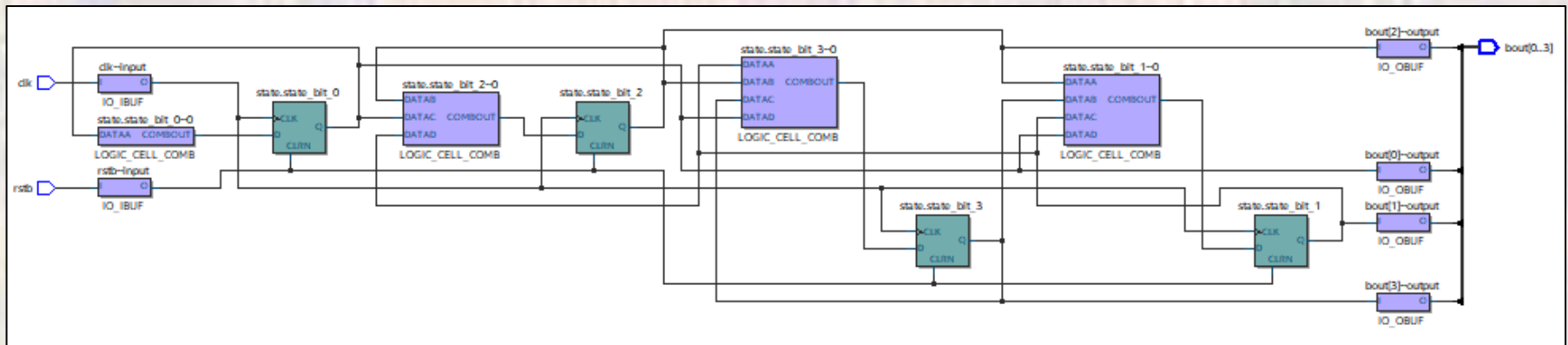
State Machine Encoding - HDL

- Encoding – Mod 10 counter

Binary

Total logic elements
Total registers

5 / 49,760 (< 1 %)
4



State Machine Encoding - HDL

- Encoding

- Comparison – Mod 10 counter
10 values to encode

- One hot (cold)

000000000**0**, 000000001**1**, 000000010**1**, 000000100**1**, 000001000**1**
000010000**1**, 000100000**1**, 001000000**1**, 010000000**1**, 100000000**1**

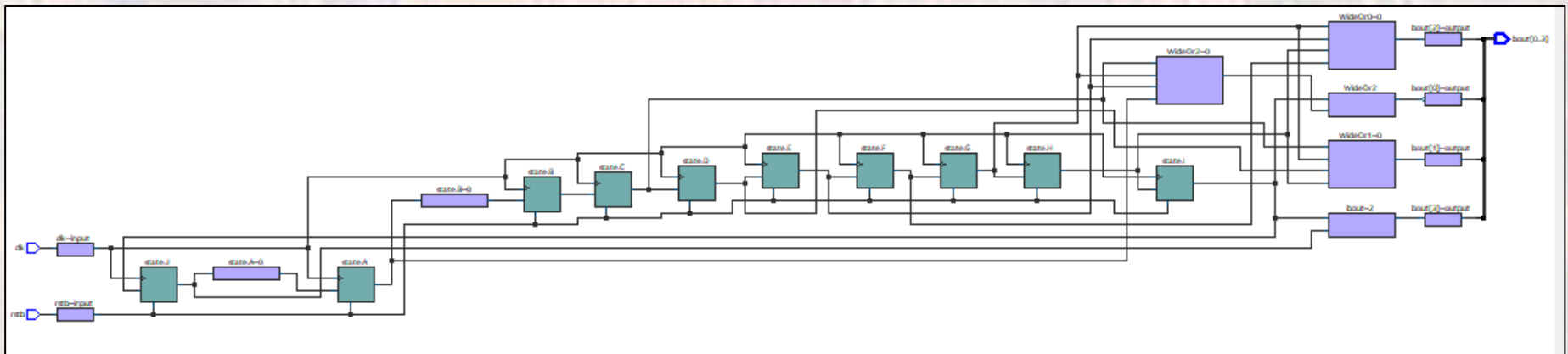
- 10 registers
- trivial decoding
 - faster
 - small
- dual bit transitions

State Machine Encoding - HDL

- Encoding – Mod 10 counter

1-Hot

Total logic elements 12 / 49,760 (< 1 %)
Total registers 10



State Machine Encoding - HDL

- Encoding
 - Comparison – Mod 10 counter
10 values to encode
- Gray
0000, 0001, 0011, 0010, 0110, 0111, 0101, 0100, 1100, 1101
- 4 registers
- complex decoding
 - slower
 - larger
- single bit transitions

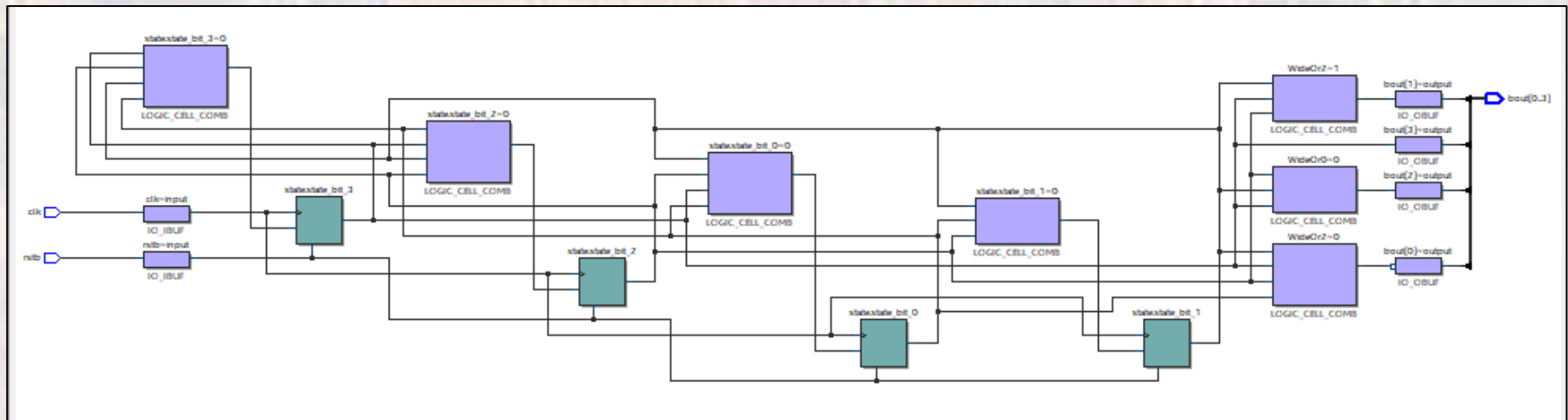
State Machine Encoding - HDL

- Encoding – Mod 10 counter

Gray

Total logic elements
Total registers

8 / 49,760 (< 1 %)
4



State Machine Encoding - HDL

- Encoding
 - Comparison – Mod 10 counter
10 values to encode
 - Johnson
00000, 10000, 11000, 11100, 11110, 11111, 01111, 00111, 00011,
00001
 - 5 registers
 - complex decoding
 - slower
 - bigger
 - single bit transitions

State Machine Encoding - HDL

- Encoding – Mod 10 counter

Johnson

Total logic elements
Total registers

15 / 49,760 (< 1 %)
5

