Автомат {#isMiliFsm}**Мили**{/isMiliFsm}{^isMiliFsm}**Мура**{/isMiliFsm}

{#isUnitaryAlgorithm}Унитарный способ кодирования {/}{#isFrequencyAlgorithm}Частотный алгоритм кодирования {/}{#isNStateAlgorithm}Способ кодирования по номеру состояния{/}{#isCanonicalAlgorithm}Канонический алгоритм кодирования{/}

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| *h* |  | *k()* |  | *k()* |  |  |  |
| {#tableData}{id} |  | {srcStateCode} |  | {distStateCode} | {#unconditionalTransition}1{/unconditionalTransition}{^unconditionalTransition}{#conditionalSignals}{#inverted}{/}{^inverted}{/}{/conditionalSignals}{/unconditionalTransition} | {#outputSignalsIndexes.length < 1}-{/}{#outputSignalsIndexes}{^isLastItem}, {/}{/outputSignalsIndexes} | {triggerExcitationSignals}{/tableData} |

Функции выходов:

{#outputFunctions}

= {#dnfEquation.terms}{#operands}{#inverted}{/inverted}{^inverted}{/inverted}{/operands}{^isLastItem} ∨ {/isLastItem}{/dnfEquation.terms} = {#shefferEquation.terms}{#operands.length > 1}({/}{#operands}{#isLogicalOperand}{symbol}{/isLogicalOperand}{^isLogicalOperand}{#inverted}{/inverted}{^inverted}{/inverted}{/isLogicalOperand}{^isLastItem} | {/isLastItem}{/operands}{#operands.length > 1}){/}{^isLastItem} | {/isLastItem}{/shefferEquation.terms}

{/outputFunctions}

Функции переходов:

{#excitationFunctions}

= {#dnfEquation.terms}{#operands}{#inverted}{/inverted}{^inverted}{/inverted}{/operands}{^isLastItem} ∨ {/isLastItem}{/dnfEquation.terms} = {#shefferEquation.terms}{#operands.length > 1}({/}{#operands}{#isLogicalOperand}{symbol}{/isLogicalOperand}{^isLogicalOperand}{#inverted}{/inverted}{^inverted}{/inverted}{/isLogicalOperand}{^isLastItem} | {/isLastItem}{/operands}{#operands.length > 1}){/}{^isLastItem} | {/isLastItem}{/shefferEquation.terms}

{/excitationFunctions}