|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Состояние | State0 | State1 | State2 | State3 |
| Двоичный код состояния S(1),S(0) | 00 | 01 | 11 | 10 |
| CNT | 0 | 1 | 1 | 0 |
| DLY\_EN | 0 | 1 | 0 | 1 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| COUNT | DLY\_OVF | S1(t) | S0(t) | S1(t+1) | S0(t+1) | SN(1) | SN(0) | Описание события |
| 0 | X | 0 | 0 | 0 | 0 | 0 | 0 | Ожидание нажатия кнопки |
| 1 | X | 0 | 0 | 0 | 1 | 0 | 1 | Нажатие кнопки |
| X | 0 | 0 | 1 | 0 | 1 | 0 | 1 | Ожидание окончания счета |
| X | 1 | 0 | 1 | 1 | 1 | 1 | 1 | Окончание счета |
| 1 | X | 1 | 1 | 1 | 1 | 1 | 1 | Ожидание отпускания |
| 0 | X | 1 | 1 | 1 | 0 | 1 | 0 | Отпускание кнопки |
| X | 0 | 1 | 0 | 1 | 0 | 1 | 0 | Ожидание окончания счета |
| x | 1 | 1 | 0 | 0 | 0 | 0 | 0 | Конец счета |

Карта Карно для SN(1)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S1,S0  COUNT,DLY\_OVF | 00 | 01 | 11 | 10 |
| 00 | 0 | 0 | **1** | **1** |
| 01 | 0 | **1** | **1** | 0 |
| 11 | 0 | **1** | **1** | 0 |
| 10 | 0 | 0 | **1** | **1** |

SN(1) = (S1 \* S0) v (S0 \* DLY\_OVF) v (S1 \* )

Карта Карно для SN(0)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S1,S0  COUNT,DLY\_OVF | 00 | 01 | 11 | 10 |
| 00 | 0 | **1** | 0 | 0 |
| 01 | 0 | **1** | 0 | 0 |
| 11 | **1** | **1** | **1** | 0 |
| 10 | **1** | **1** | **1** | 0 |

SN(0) = ( \* S0) v ( \* COUNT) v (S0 \* COUNT)

CNT = S(0)

DLY\_EN = S1 xor S0

Описание устройства:

LIBRARY ieee;

USE ieee.std\_logic\_1164.ALL;

USE IEEE.STD\_LOGIC\_ARITH.ALL;

USE IEEE.STD\_LOGIC\_UNSIGNED.ALL;

Entity lab2\_example IS

PORT (

RST: IN STD\_LOGIC; --Системный сигнал сброса

CLK: IN STD\_LOGIC; --Сигнал синхронизации

COUNT: IN STD\_LOGIC; --Сигнал кнопки с дребезгом

CNT: OUT STD\_LOGIC --Сигнал кнопки, очищенный от дребезга

);

END lab2\_example;

ARCHITECTURE behavioral OF lab2\_example IS

-- Кодируем состояния в соответствии с вариантом

CONSTANT STATE0: STD\_LOGIC\_VECTOR (1 downto 0) := "00";

CONSTANT STATE1: STD\_LOGIC\_VECTOR (1 downto 0) := "01";

CONSTANT STATE2: STD\_LOGIC\_VECTOR (1 downto 0) := "11";

CONSTANT STATE3: STD\_LOGIC\_VECTOR (1 downto 0) := "10";

-- Состояние автомата в момент времени t

SIGNAL S: STD\_LOGIC\_VECTOR (1 downto 0);

-- Состояние автомата в момент времени t+1

SIGNAL SN: STD\_LOGIC\_VECTOR (1 downto 0);

SIGNAL COUNTER: integer; -- Счетчик 2^20

SIGNAL DLY\_OVF: STD\_LOGIC; -- Сигнал "Завершение счета"

SIGNAL DLY\_EN: STD\_LOGIC; -- Сигнал разрешения работы счетчика

BEGIN

-- Память состояний

FSM\_STATE\_inst: PROCESS (CLK)

BEGIN

IF (CLK='1' and CLK'event) THEN

IF (RST='1') THEN

S <= STATE0;

ELSE

S <= SN;

END IF;

END IF;

END PROCESS;

-- Комбинационная схема для выработки сигналов CNT и DLY\_EN (по индивидуальному варианту)

CNT <= S(0);

DLY\_EN <= S(1) xor S(0);

--Комбинационные схемы для определения следующего состояния (по индивидуальному варианту)

SN(0) <= (NOT S(1) AND S(0)) OR (NOT S(1) AND COUNT) OR (S(0) AND COUNT);--пример 1

SN(1) <= (S(1) AND S(0)) OR (S(0) AND DLY\_OVF) OR (S(1) AND NOT DLY\_OVF);--пример 2

-- Описание счетчика

COUNTER\_inst: PROCESS (CLK)

BEGIN

IF (CLK='1' and CLK'event) THEN

IF (RST='1' or DLY\_EN = '0') THEN

COUNTER <= 0;

ELSE

COUNTER <= COUNTER + 1;

END IF;

END IF;

END PROCESS;

DLY\_OVF <= '1' WHEN COUNTER = 2\*\*7-1 ELSE '0'; --Длительность задержки

END Behavioral;

Скриншоты



