

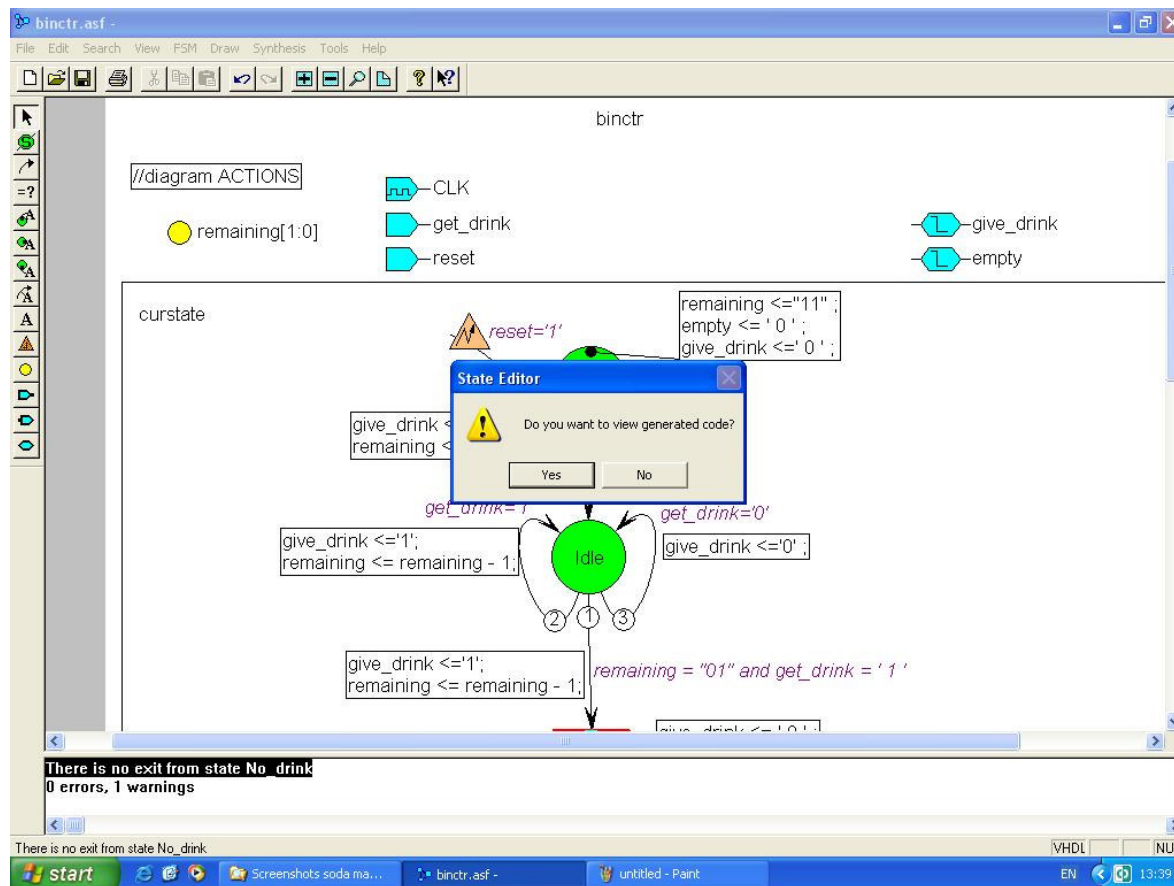
Пример 2: Последователност на работа с графичен редактор ACTIVE-HDL FSM за въвеждане на спецификация на схема чрез крайни автомати (FSM)

Автори: Мария Дамянова, Галя Маринова

## 2.2.) Генериране на VHDL код от спецификацията на автомата FSM

=> Изберете **Synthesis -> HDL Code Generation**.

**Забележка** – Ще забележите едно предупреждение, че няма изходно състояние от състоянието No\_drink. Можете да игнорирате това предупреждение.



Фигура 12 Генериране на VHDL кода.

=> За да видите Вашия генериран VHDL код, натиснете **Yes** в диалоговия прозорец, който се появява след генерирането на кода. Така създавате binctr.vhd, който може да бъде използван вместо ръчно въведен binctr.vhd файл. Вижте финалния VHDL код, изложен отдолу.

След като тествате VHDL файл-а със симулатора, пробвайте да смените приоритетите на условията. Погледнете генерирания код и вижте как се е променил.

**Пример 2: Последователност на работа с графичен редактор ACTIVE-HDL FSM за въвеждане на спецификация на схема чрез крайни автомати (FSM)**

**Автори: Мария Дамянова, Галя Маринова**

```
--
-- File: C:\Mariya Damyanova\soda machine s HDL FSM\binctr.vhd
-- created: 04/24/09 13:39:28
-- from: 'C:\Mariya Damyanova\soda machine s HDL FSM\binctr.asf'
-- by fsm2hdl - version: 2.0.1.45
--
library IEEE;
use IEEE.std_logic_1164.all;

package binctr_pkg is
    component binctr
        port (reset, get_drink, clk: in std_logic;
            give_drink: inout std_logic;
            empty: inout std_logic) ;
    end component;
end binctr_pkg;

library IEEE;
use IEEE.std_logic_1164.all;

library CYPRESS;
use CYPRESS.std_arith.all;
use CYPRESS.lpm_pkg.all;
entity binctr is
    port (CLK: in STD_LOGIC;
        get_drink: in STD_LOGIC;
        reset: in STD_LOGIC;
        empty: inout STD_LOGIC;
        give_drink: inout STD_LOGIC);
end;

architecture binctr_arch of binctr is

--diagram signal declarations
signal remaining: STD_LOGIC_VECTOR (1 downto 0);

-- SYMBOLIC ENCODED state machine: curstate
type curstate_type is (Idle, No_drink, Start);
signal curstate: curstate_type;

begin
--concurrent signal assignments
--diagram ACTIONS;

curstate_machine: process (CLK, reset)
begin
if reset='1' then
    remaining <="11" ;
    empty <= ' 0 ' ;
    give_drink <=' 0 ' ;
    curstate <= Start;
elsif CLK'event and CLK = '1' then
    case curstate is
        when Idle =>
            if remaining = "01" and get_drink = ' 1 ' then
                curstate <= No_drink;
                give_drink <='1';
            end if;
        end case;
    end process;
end architecture;
```

**Пример 2: Последователност на работа с графичен редактор ACTIVE-HDL FSM за въвеждане на спецификация на схема чрез крайни автомати (FSM)**

**Автори: Мария Дамянова, Галя Маринова**

```
                remaining <= remaining - 1;
            elsif get_drink='1' then
                curstate <= Idle;
                give_drink <='1';
                remaining <= remaining - 1;
            elsif get_drink='0' then
                curstate <= Idle;
                give_drink <='0' ;
            end if;
        when No_drink =>
            give_drink <= ' 0 ' ;
            empty <= ' 1 ' ;
        when Start =>
            if get_drink = ' 1 ' then
                curstate <= Idle;
                give_drink <='1';
                remaining <= remaining - 1;
            end if;
        when others =>
            null;
    end case;
end if;
end process;

end binctr_arch;
```

Следва компилиране и синтезиране на binctr и попълване на VHDL описанията в CY7C37256 JEDEC файл.