

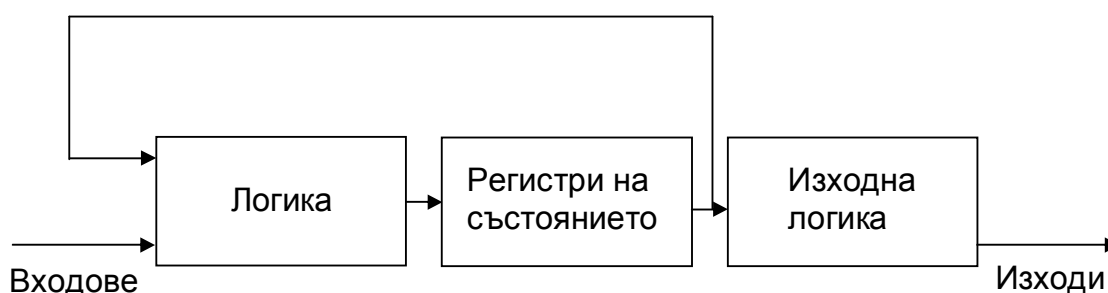
Пример 2: Последователност на работа с графичен редактор ACTIVE-HDL FSM за въвеждане на спецификация на схема чрез крайни автомати (FSM)  
Автори: Мария Дамянова, Галя Маринова

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

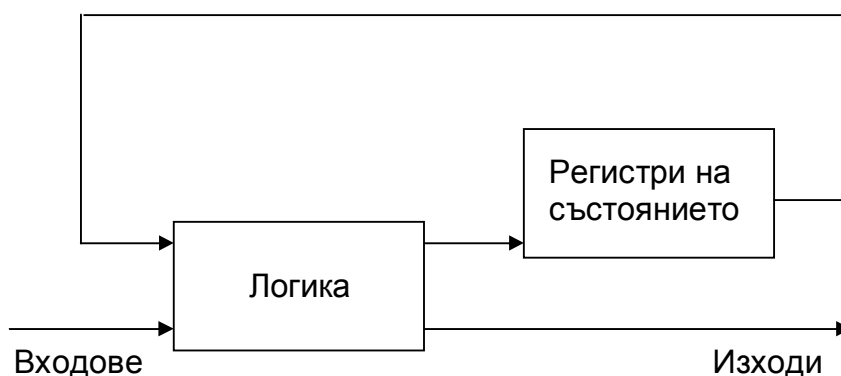
Работата на графичния редактор **ACTIVE-HDL FSM** е описана чрез пример за създаване на автомат за продажба на сода. Автоматът е създаден със схема от фамилията Ultra37000.

Програмата позволява да се генерира автоматично VHDL код от спецификацията с крайни автомати, след което се преминава към симулация на автомата в симулатора **ACTIVE-HDL Sim**.

Съществуват два типа крайни автомати – на Мур и на Мели, представени на фигура 1 и фигура 2.



Фигура 1 Диаграма с краен автомат на Мур



Фигура 2 Диаграма с краен автомат на Мели

При модела на Мур резултатът зависи само от входовете, докато при модела на Мели във формирането на резултата участва и информацията за предишните състояния.

## Създаване на спецификация на автомат за продажба на сода в ACTIVE-HDL FSM

Целта на това ръководство е да демонстрира основните характеристики на графичния редактор **Active-HDL FSM**. Автоматът, който се създава, ще генерира VHDL файл, **binctr.vhd**, който може да се използва вместо ръчно

**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**

създаден **binctr.vhd** файл. Времето, необходимо за създаване на VHDL файловете по двата начина, е съизмеримо.

Съществуват различни начини да се създаде този автомат и след това могат да се сравнят алтернативни решения. Някои възможности са предложени в самото ръководството.

### **Описание на проекта.**

В този пример Вие ще проектирате регулатор за машина за напитки. Машината има отделение, за напитки. Отделението съдържа три напитки. (Това може да бъде произволна стойност, но три е лесно за симулация число.)

Автоматът освобождава напитка, ако потребителят натисне бутона за напитката и поне един напитка е налична. Появява се сигнал за напълване (**empty signal**), ако липсват две напитки. Натискането на сигнала **reset** казва на схемата, че машината е заредена и отделенията са пълни.

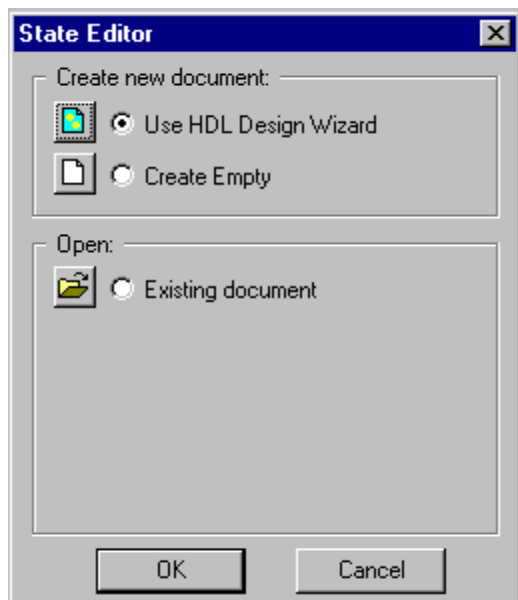
Пример 2: Последователност на работа с графичен редактор ACTIVE-HDL FSM за въвеждане на спецификация на схема чрез крайни автомати (FSM)

Автори: Мария Дамянова, Галя Маринова

## 2.1.) Създаване на проект на автомат в графичния редактор ACTIVE-HDL FSM във файл `binctr.asf`.

За да стартирате Active-HDL FSM докато сте в текстовия редактор **Galaxy**, изберете **Tools -> Active-HDL FSM**.

Когато отворите Active-HDL FSM, ще видите диалоговия прозорец от фигура 1.



Фигура 1 The State Editor диалогов прозорец

=> Изберете **HDL Design Wizard**, ако не е избрано вече.

=> Натиснете **OK**.

=> Натиснете **Next**.

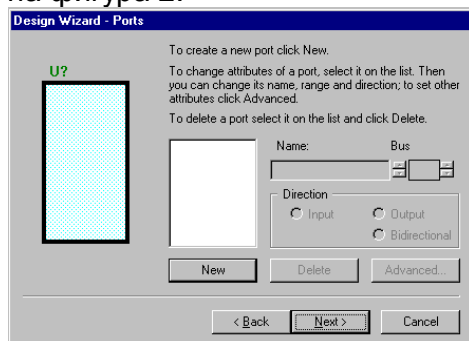
=> Изберете **VHDL** като Ваш предпочитан език.

=> Натиснете **Next**.

=> Натиснете бутона **browse**, за да сте сигурни, че сте във правилната директория, `c:\your name`.

=> Въведете името на файла като **binctr**. Така ще създадете файл на име **binctr.asf**.

=> Натиснете **Next**. Диалоговият прозорец, който ще видите, ще изглежда като на фигура 2.



Фигура 2 The Design Wizard - Ports – диалогов прозорец преди въвеждане портовете

## 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

Сега Вие ще създадете портовете за binctr. Ще създадете два входа и два изхода.

=> Натиснете **New**. Въведете **reset** в полето Name и се уверете, че бутонът **Input** е избран.

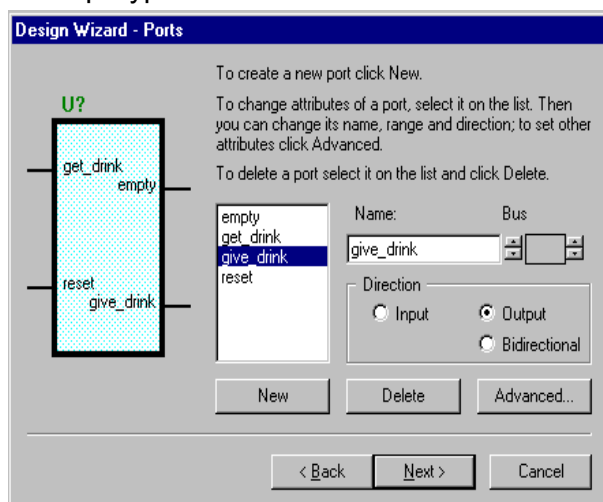
=> Натиснете **New**. Въведете **get\_drink** в полето Name и се уверете, че бутонът **Input** е избран.

=> Натиснете **New**. Въведете **empty** в полето Name и се уверете, че бутонът **Bidirectional** е избран.

=> Натиснете **Advanced** и се уверете, че **Registered mode** е избран.

=> Натиснете **New**. Въведете **give\_drink** в полето **Name** и се уверете, че бутонът **Bidirectional** е избран.

След като добавите всички входове и изходи, прозорецът трябва да изглежда като фигура 3.



=> Натиснете **Next**.

**Забележка** - Design Wizard ще Ви предупреди, че не сте въвели порт на име CLK. Натиснете Yes, така че Design Wizard да създаде порт за тактовия генератор на име CLK.

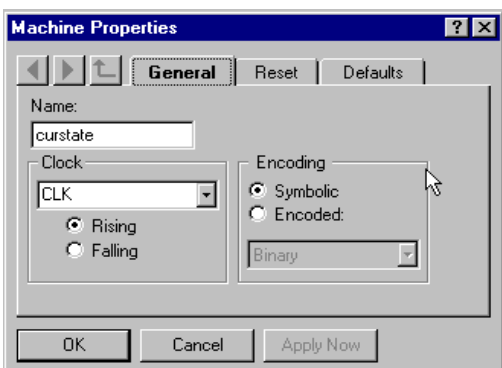
=> Следващият диалогов прозорец пита колко автомата ще създадете. Изберете One и натиснете Finish.

=> Можете да увеличите или да намалите мащаба, като изберете **View -> Zoom in** или **View -> Zoom out** от менюто на Active-HDL FSM.

Фигура 3 Диалогов прозорец The Design Wizard - Ports след въвеждане портовете

### 2.1.1. Създаване на автомат curstate

По подразбиране името на автомата е **Sreg0** и се намира в горния ляв ъгъл на автомата. Има две полета: полето на интерфейса с портове и символи и полето на автомата.



=> Натиснете два пъти върху текста „Sreg0“ с левия бутон на мишката. Изтрийте Sreg0 и въведете **curstate**. Натиснете **Enter/Return**.

=> Натиснете с десен бутон някъде в машината и изберете Properties от pop-up менюто. Диалоговият прозорец Machine Properties, показан на фигура 4, е мястото, където ще нагласите свойствата на автомата.

Фигура 4 Диалогов прозорец Machine Properties.

=> При таба **General**, настройките по подразбиране трябва да са **Rising** и **Symbolic**.

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

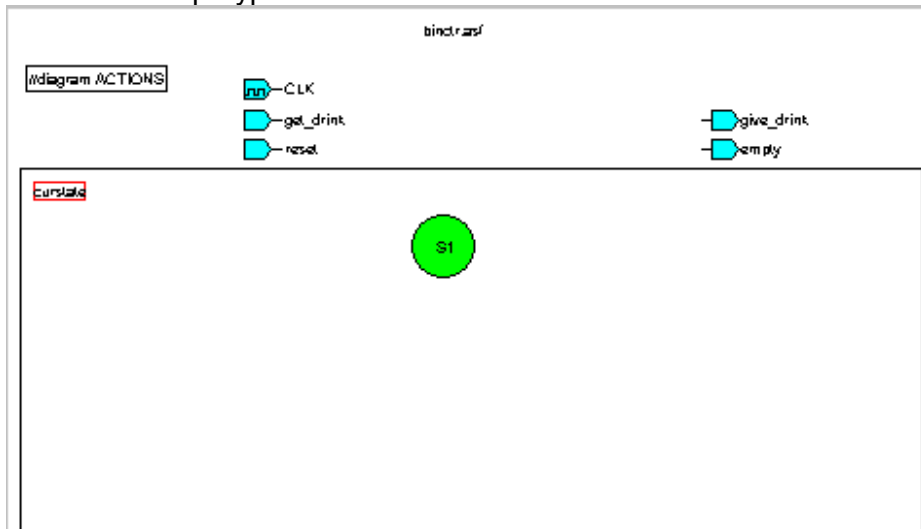
Автори: Мария Дамянова, Галя Маринова

=> Изберете таба **Reset**. Настройките по подразбиране трябва да са **Asynchronous** и **High**.

=> Натиснете **OK**, за да излезете от този диалогов прозорец.

=> Натиснете бутона **state** от FSM toolbar-a.

=> Курсорът на мишката ще е символ за състояние. Поставете го както е показано на фигура 5.



Фигура 5 Поставяне на символа за автомат

=> Натиснете два пъти върху текста S1. Въведете следния текст и натиснете бутона enter/return:

**Start**

=> Натиснете с десен бутон в диаграмата и изберете Properties от pull-down менюто.

Първото състояние в автомата curstate ще бъде състоянието **reset**. Този набор от инструкции асоциира порта **reset** като начално състояние.

=> Натиснете таба **Reset**.

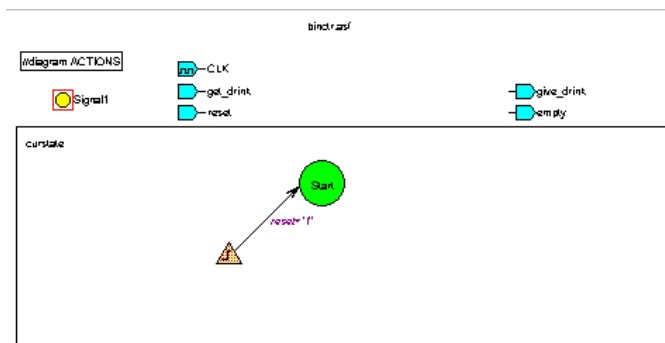
=> Натиснете pull-down менюто за полето **Port Name**. Изберете **reset**.

=> Натиснете pull-down менюто за полето **State**. Изберете **Start**.

=> Натиснете **OK**, за да се върнете към диаграмата.

Сега трябва да създадем сигнал, наречен **remaining**, за да бъде използван като брояч колко кутийки напитки са останали.

=> Натиснете бутона **signal** от FSM toolbar-a.



Фигура 6 Поставяне на сигнала.

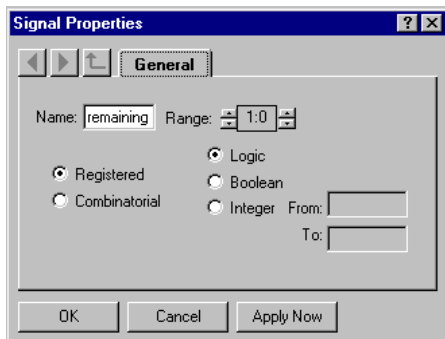
=> Натиснете извън диаграмата от лявата страна на входните портове и поставете сигнала, както е показано на фигура 6. Така създавате VHDL сигнал. Ако бяхме сложили сигнала вътре в автомата, това щеше да бъде VHDL променлива.

## 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

=> Натиснете два пъти, за да редактирате текста, въведете следното и натиснете **enter/return**:

**remaining**



=> Натиснете с десен бутон върху сигнала (не името) и изберете Properties.

=> Натиснете върху горния ляв range бутон един път, за да настроите ранга като 1:0. Уверете се, че е избрано **Registered**. Диалоговият прозорец Signal Properties, показан на фигура 7, е мястото, където ще нагласите свойствата на сигнала **remaining**.

Фигура 7 Диалогов прозорец Signal Properties за настройване на ранга на сигнала.

=> Натиснете **OK**.

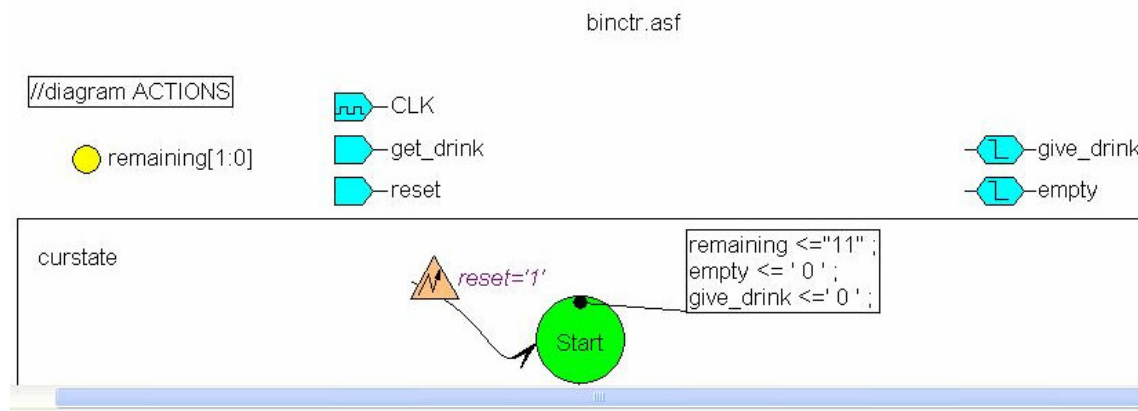
=> Натиснете бутона **entry action** от FSM toolbar-a.

=> Поставете точката в символа на началното състояние **Start** и въведете следния текст:

**remaining <="11";**

**empty <= '0';**

**give\_drink <='0';**



Фигура 8 Въвеждане на текст в началното състояние Start.

=> Създайте ново състояние под символа на състоянието **Start**.

=> Натиснете два пъти, за да редактирате текста, въведете следния текст и натиснете enter/return:

**Idle**

=> Натиснете бутона за преход **transition** от FSM toolbar-a.

=> Натиснете символа на състоянието **Start**, издърпайте курсора на мишката надолу до състоянието **Idle** и натиснете отново, за да настроите прехода.

**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**

**Забележка** – Когато правите стрелка за преход, можете да натиснете с левия бутон на мишката на няколко точки в бялото поле, за да направите дъгата на стрелката, така както желаете.

Сега създадохме състоянието **idle**. Ще направим преход от състоянието **start** към **idle**, ако някой поиска питие.

=> Натиснете бутона **condition** (условие) от FSM toolbar-a, поставете курсора върху стрелката за преход от Start към Idle, натиснете и въведете следния текст:

```
=> get_drink = '1'
```

=> Натиснете бутона **transition action** от FSM toolbar-a, поставете курсора върху стрелката за преход от **Start** към **Idle**, натиснете и въведете следния текст:

```
give_drink <= '1';  
remaining <= remaining - 1;
```

**Забележка** – Вашият текстов прозорец може да е върху условието. Когато сте въвели текста, просто натиснете текстовия прозорец и влачете мишката, за да го преместите.

**Забележка** – Когато въвеждаме действия, ние завършваме VHDL кода с „;“.

Когато въвеждаме условия, не използваме точка и запетая. Това е така, защото текстът, който въвеждате като условие или действие, ще се превърне в истински VHDL код, който ще бъде създаден по-късно.

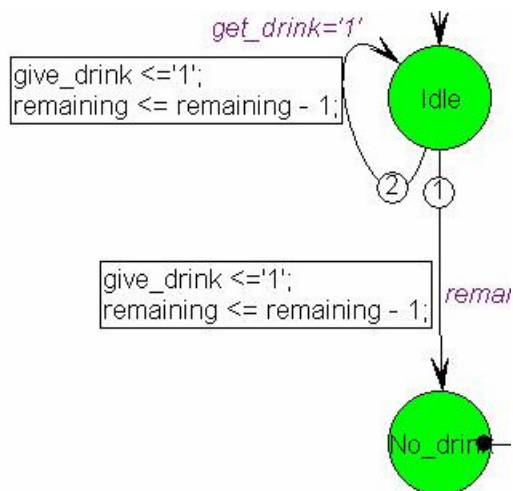
=> Създайте ново състояние под символа на състоянието **state**.

=> Натиснете два пъти, за да редактирате текста, въведете следния текст и натиснете enter/return:

**No\_drink**

=> Натиснете бутона state action от FSM toolbar-a (лента с инструкциите).

=> Поставете точката върху символа **No\_drink**, въведете следния текст и натиснете enter/return:



```
=> give_drink <= '0';
```

```
=> empty <= '1';
```

=> Натиснете бутона **transition** от FSM toolbar-a.

=> Натиснете символа на състоянието **Idle**, издърпайте курсора надолу към състоянието No\_drink и натиснете отново, за да настроите прехода.

Сега ще добавим условия и действия за напускане на състоянието **Idle**.

**Първо**, ако е останало само едно питие и някой поиска питие.

Фигура 9 Въвеждане на текст за прехода

**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**

=> Натиснете бутона **condition** от FSM toolbar-a, поставете курсора върху стрелката за преход от **Idle** към **No\_drink**, натиснете и въведете следния текст:  
remaining = „01“ and get\_drink = '1'

=> Натиснете бутона **transition action** от FSM toolbar-a, поставете курсора върху стрелката за преход от **Idle** към **No\_drink**, натиснете и въведете следния текст:

```
give_drink <= '1';  
remaining <= remaining – 1;
```

между състоянията **Start** и **No\_drink**.

=> Натиснете с десен бутон върху стрелката за преход от **Idle** към **No\_drink**, изберете **Priority** от менюто, изберете 1 от предложените числа и натиснете с ляв бутон на мишката.

Натиснете в автомата, за да настроите приоритет.

**Забележка** - Приоритетите влияят на реда на условията, отразено в генерирания на VHDL финален код.

**Второ**, ако някой поиска питие, дай питието и се върни към състоянието.

=> Натиснете бутона **transition** от FSM toolbar-a.

=> Натиснете отдолу на символа на състоянието **Idle**, издърпайте курсора наляво в бялата част на диаграмата на автомата, натиснете с ляв бутон на мишката и издърпайте курсора нагоре към състоянието **Idle state** и натиснете отново, за да настроите прехода. За помощ относно поставянето погледнете фигура 8.

=> Натиснете бутона **condition** от FSM toolbar-a, поставете курсора върху стрелката за преход от **Idle** към **Idle**, натиснете и въведете следния текст:

```
get_drink <= '1';
```

=> Натиснете бутона **transition action** от FSM toolbar-a, поставете курсора върху стрелката за преход от **Idle** към **Idle**, натиснете и въведете следния текст:

```
give_drink <= '1';  
remaining <= remaining – 1;
```

=> Натиснете с десен бутон върху стрелката за преход от **Idle** към **Idle**, изберете **Priority** от менюто, изберете 2 от предложените числа и натиснете с ляв бутон на мишката.

Натиснете в автомата за да настроите приоритета.

**Трето**, ако няма заявки, върни се към състоянието и настрой **give\_drink** на нула.

=> Натиснете бутона **transition** от FSM toolbar-a.

=> Натиснете отдолу на символа на състоянието **Idle**, издърпайте курсора надясно в бялата част на диаграмата на автомата, натиснете с ляв бутон на мишката и издърпайте курсора нагоре към състоянието **Idle state** и натиснете отново, за да настроите прехода. За помощ относно поставянето погледнете фигура 10.



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

=> Натиснете бутона **condition** от FSM toolbar-а, поставете курсора върху стрелката за преход от **Idle** към **Idle**, натиснете и въведете следния текст:

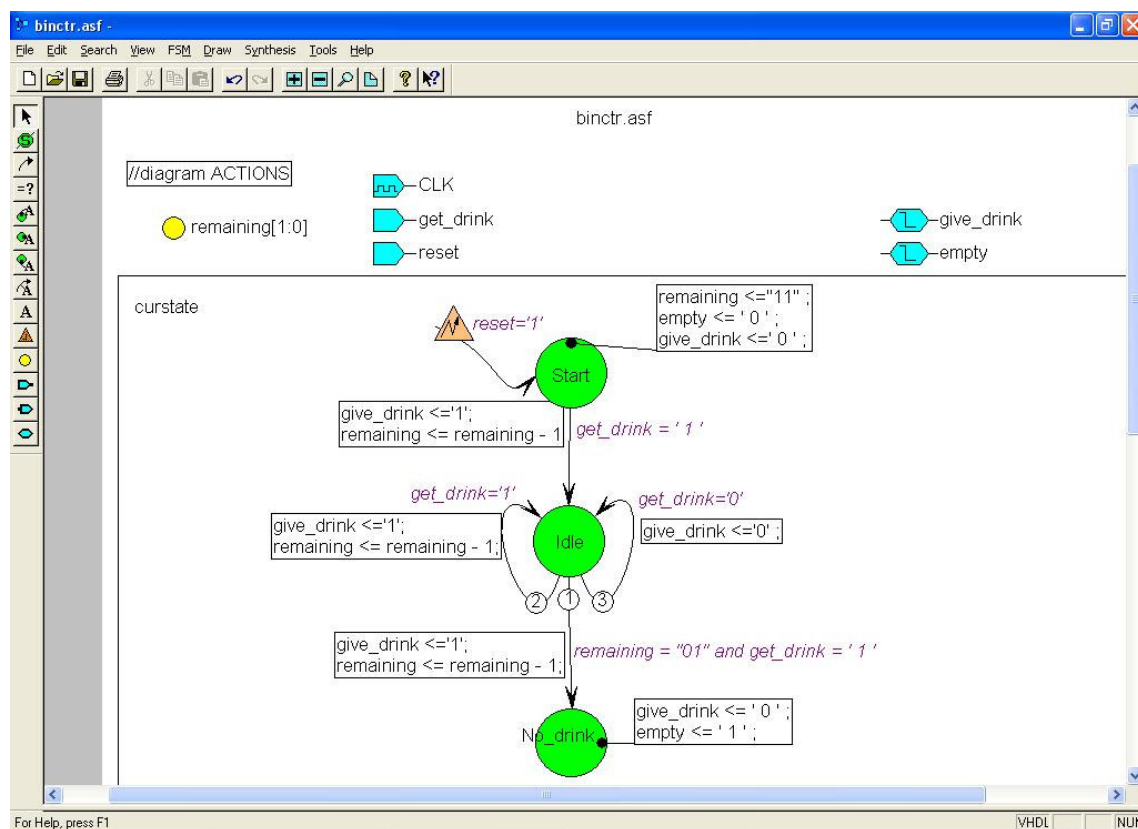
```
get_drink <= '0';
```

=> Натиснете бутона **transition action** от FSM toolbar-а, поставете курсора върху стрелката за преход от **Idle** към **Idle**, натиснете и въведете следния текст:

```
give_drink <= '0';
```

=> Натиснете с десен бутон върху стрелката за преход от **Idle** към **Idle**, изберете **Priority** от менюто, изберете 3 от предложените числа и натиснете с ляв бутон на мишката.

Натиснете в автомата за да настроите приоритета. Финалната диаграма трябва да изглежда като фигура 10.



Фигура 10 Финална диаграма на binctr.asf

## 2.1.2. Избиране на библиотеки

Изберете **Synthesis** -> Изберете **Libraries**.

⇒ Въведете следния текст СЛЕД текста по default:

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

**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**

```
        empty: inout std_logic);  
    end component;  
end binctr_pkg;
```

Декларацията на package дава информация на Warp компилатора да разреши ползването на binctr като компонент от проект за проектиране на по-високо ниво.

**Забележка** – Декларацията на package **трябва** да бъде преди entity декларацията и архитектурата във .vhd файл-а.

Първият ред от декларацията на пакета дава име на package. Името на package трябва да е различно от имената на други компоненти, деклариращи в този пакет. Използването на конвенцията <entity>\_pkg работи добре.

Вторият ред декларира компонент на име binctr. Името на компонента, което се появява на този ред трябва да съвпада с името на придружаващото го **entity**.

Твърдението за порта декларира името, посоката и типа на всеки порт в този компонент.

За тази цел можете да копирате твърдението за порта от декларацията на **entity-то**.

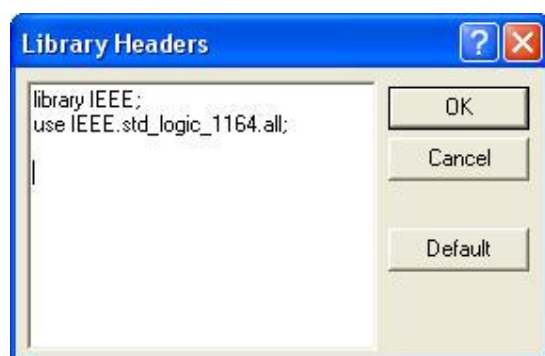
Краен компонент и крайно binctr\_pkg твърдение завършват декларацията на пакета. Обърнете внимание, пакетът, наименован в последното твърдение на пакета, трябва да съвпадне с посочения в първия ред на декларацията на пакета.

=> Въведете следния текст след декларацията на пакета.

```
library IEEE;  
use IEEE.std_logic_1164.all;
```

```
library CYPRESS;  
use CYPRESS.std_arith.all;  
use CYPRESS.lpmpkg.all;
```

=> Натиснете **OK**.



Фигура 11 Въвеждане на текст за прехода между **Start** и **No\_drink**.