ДОДАТОК А

Графічний матеріал атестаційної роботи

Харківський університет радіоелектроники

Кафедра електронних обчислювальних машин

Модуль управління імітаційної системи

Студент групи СПзм-18-2 Наумов І.А.

Керівник проф. кафедри ЕОМ, к.т.н. Горбачов В.О

Харьков 2020

Задачі дослідження:

1) провести аналіз і запропонувати найбільш ефективні математичні схеми базових моделей, з яких будуються більші моделі,

2) провести аналіз і запропонувати формальних засобів з'єднання цих моделей один з одним,

3) розробити структуру модуля управління процесом моделювання в імітаційної системі,

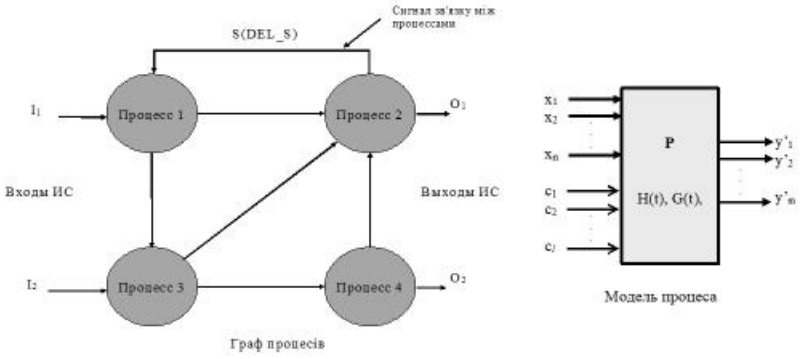
4) програмно реалізувати алгоритми: складання черги планування подій і алгоритм модифікації тимчасової координати і управління чергою подій.

Етапи створення і використання моделей

Комп'ютерне моделювання припускає наявність наступних етапів створення і використання моделей:

- складання змістовного опису об'єкта, що досліджується;
- розробка концептуальної моделі;
- розробка математичної моделі;
- калібрування і перевірка придатності моделі;
- програмування моделі;
- планування і проведення машинних експериментів;
- аналіз результатів моделювання.

Процеси, як засіб формалізації динаміки елементів складних систем



ОПЕРАТОРИ ПРОЦЕСУ

Перша група операторів управлення станом процесу:

$$z(t_2) = H_1(z(t_1), x(t_1), c(t_1));$$

wait ((список сигналів управління), (умова),(час));

Друга група операторів визначає вихід процесу:

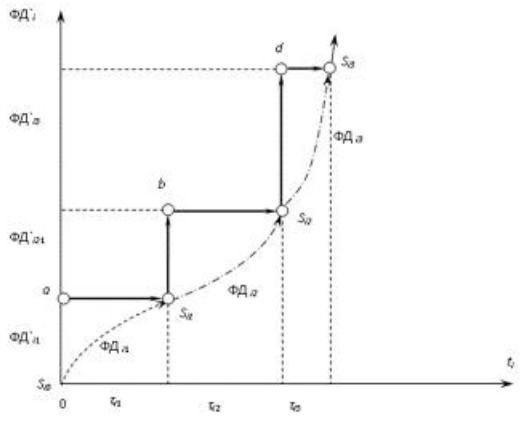
$$y(t_2) = G_1(z(t_1), x(t_1), c(t_1));$$

$$\Delta t(t_2) = G_2(z(t_1), x(t_1), c(t_1));$$

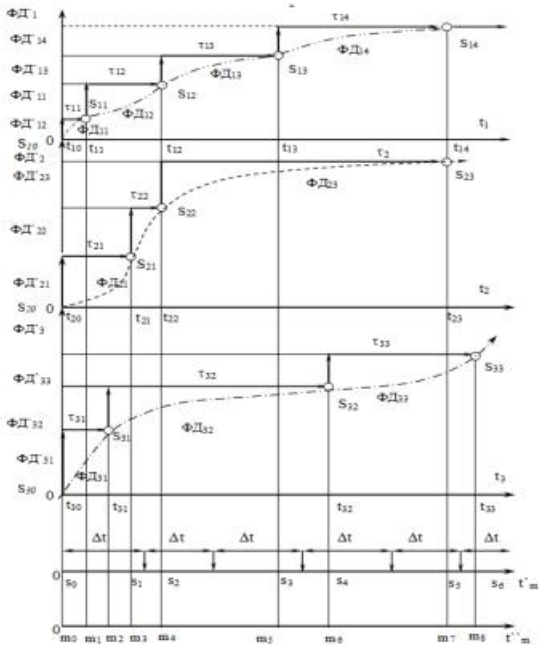
$$a(t_2) = G_3(z(t_1), x(t_1), c(t_1));$$

$$y^*(t_2) = G_4(y(t_2), \Delta t(t_2), a(t_2)),$$

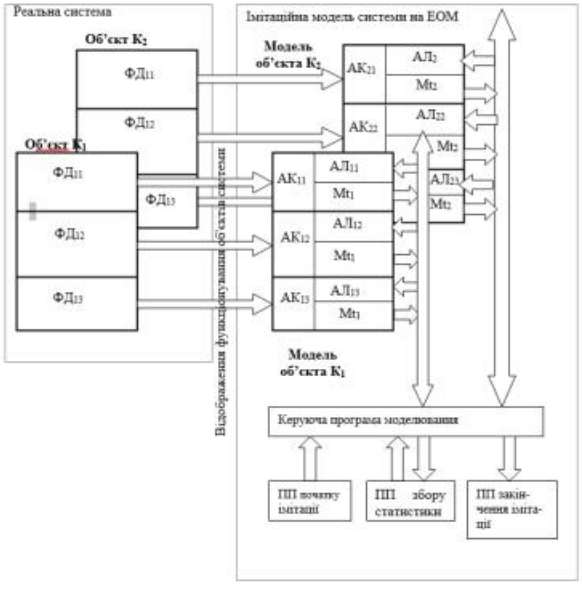
Апроксимація функціональних дій об'єкта Кі у моделі системи



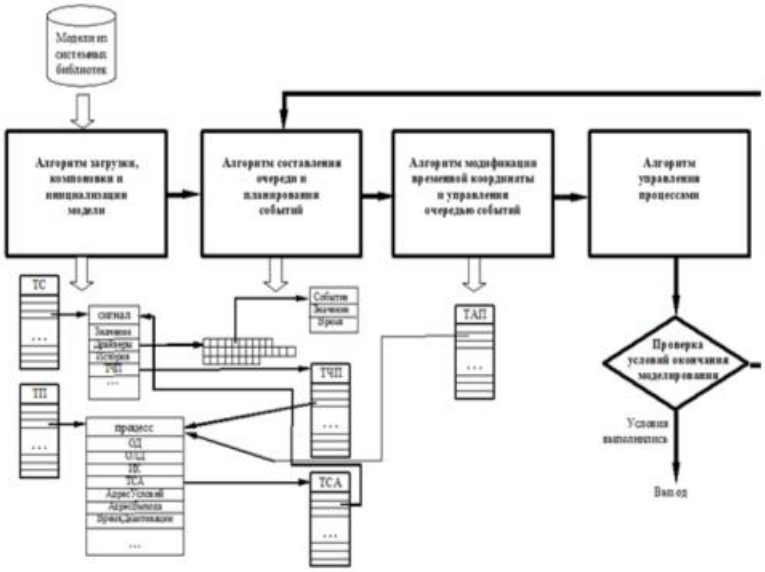
ЧАСОВА ДІАГРАМА ПОДІЙ У РЕАЛЬНІЙ СИСТЕМІ,
ЯКА СКЛАДАЄТЬСЯ З ТРЕХ КОМПОНЕНТІВ



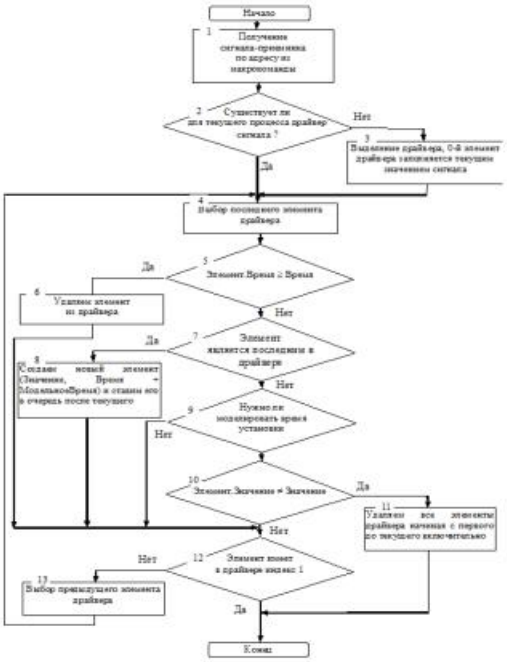
Основні компоненти імітаційних моделей імітаційних систем



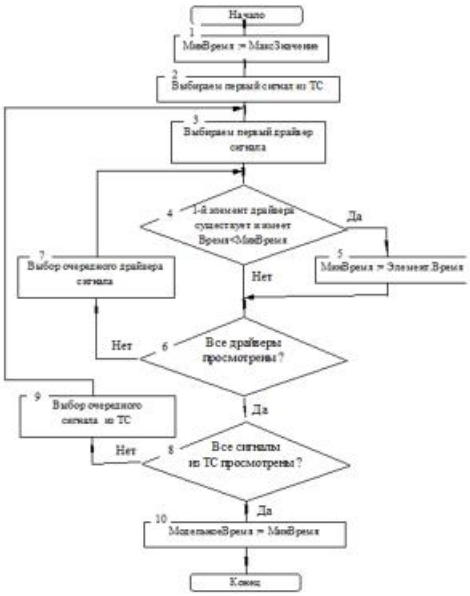
Основні алгоритми і та структури даних модуля управління



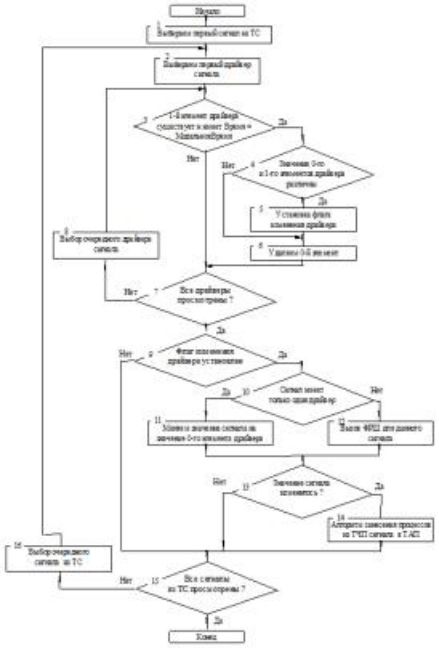
Алгоритм складання черги і планування подій



Алгоритм модифікації часової координати



Алгоритм перебудови черг подій



ВИСНОВКИ

При реалізації даного програмного комплексу моделювання були вирішені наступні завдання:

- розглянуті характеристики електронних компонентів і запропоновані вимоги моделей системного рівня;
- розроблена процесна модель елемента складної електронної системи;
- розроблена структура і алгоритм з використанням процесного підходу;
- програмно реалізований алгоритми: складання черги планування подій і алгоритм модифікації тимчасової координати і управління чергою подій;

ДОДАТОК Б

Код розробленої програми

```
library ieee;
use ieee.std_logic_1164.all;
use std.textio.all;
library pro_lib;
use pro_lib.TYPESandFUNC.all;
use work.all;
entity TESTBNCH is
end TESTBNCH;
architecture stimulus of TESTBNCH is
component PIPELINE is
    generic ( twoPERIOD : TIME );
    port ( CLK: in bit;INTR,READY: in std_ulogic;
           INTA: out  std_ulogic; RESET:in std_ulogic;
            DATA: inout WORD_TSL;
            ADDRESS: out ADDRESS24;
            RD, WR: inout std_ulogic;
            ReadyCON: in std_ulogic;
            SIZE: inout INTEGER;
            SizeOut  :in integer;
            BC_BUFF: inout DB;
            PERIPHERAL_RG :inout BYTE_ARR (7936 to 8191)
         );
end component;
component INTCLOCK is
    generic ( twoPERIOD : TIME );
    port ( XTAL1: in bit;
           CLKOUT: out bit;
           CLKOUT_4: out bit  );
end component;ï
```

Приклад Б.1 – Код програми складання черги планування подій

```
------- BUS Controler -----
component BUS_CONTR

  port( BW : inout bit;CLKOUT: in bit;
      C_Size      :inout integer; READY, Reset,
        EA : in std_ulogic;
        BUSWIDTH :in bit;
        WRH_BHE :inout bit;
        C_BHE :inout bit;
        WRL_WR  :inout bit;
        RD :inout bit;
        INST :out bit;
        C_READ  ,C_WRITE,
        ALE_ADV :inout bit;
```

Приклад Б.2 – Продовження

```
library ieee;
use ieee.std_logic_1164.all;
use std.textio.all;
use work.all;

library pro_lib;
use pro_lib.TYPESandFUNC.all;
entity PIPELINE is
    generic (twoPERIOD : TIME );
    port (
        CLK: in bit;INTR, READY: in std_ulogic;
        INTA: out  std_ulogic;
        Reset:in std_ulogic;
         DATA: inout DATA16;
         ADDRESS: out ADDRESS24;
         RD, WR: inout std_ulogic;
         ReadyCON: in std_ulogic;
         SIZE: inout INTEGER;
         SizeOut  :in integer;
         BC_BUFF: inout DB;
         PERIPHERAL_RG :out BYTE_ARR (7936 to 8191)
      );
end PIPELINE;

architecture BEHAVIOR_PIPELINE of PIPELINE is

 component FILE_DEV is
    port (
            RESET: in std_ulogic;
        INTERNAL_CODE_RAM :in BYTE_ARR (0 to 2043);
        REGISTER_FILE :in BYTE_ARR(0 to 511)
        );
 end component;

  constant QUEUE_SIZE : integer := 10;
  constant COMM_RG_SIZE : integer := 7;
  constant CLK_INTERRUPT: integer := 4;
```

Приклад Б.3 – Код програми модифікції часової координати та керування чергою подій

```
     signal  FETCH,  DECODE,  READ_EXEC,  READ_EXECUTE,  WRITE,  EXECUTE_WRITE,
READ, clk_rd3 : BOOLEAN := false;
       signal clk_w1,clk_r2,clk_w2,clk_r3,clk_w3 : integer := 0;
       signal       CLK_FT1,CLK_FT2,CLK_DEC1,CLK_DEC2,CLK_RE1,CLK_RE2,CLK_EW1,
CLK_EW2,CLK_RES: bit;

       signal COMM_LEN_DEC : integer := 0;
```

Приклад Б.4 – Продовження