Kávéautomata vezérlőjének megvalósítása magas szintű szintézist alkalmazva.

```
#define KESZENLET 0
#define ELOKESZIT 1
#define VIZET_MELEGIT 2
#define KAVET_ADAGOL 4
#define CUKROT_ADAGOL 8
#define TEJET_ADAGOL 16
#define VIZET_ADAGOL 32
#define KEVERI 64
#define KESZ 128

#include "kave_automata.h"

 static unsigned int szamlalo=25;
 static allapot_tipus allapot=RDY;
 void kaveautomata(din_t T_MATRIX[N], dout_t  *VEZERLO_JELEK, din_t rst, din_t start, din_t pf)
{
  din_t T_VAL_MAT[N]; //belső változó: a modulon belül tároljuk bemásoljuk az időzítésekhez rendelt
értékeket
  dout_t VEZERLO_VAL; //belső változó: tároljuk a vezérlőjelek pillanatnyi állapotát
  unsigned char i;

  //bemásoljuk az időzítésekhez rendelt értékeket a  T_VAL_MAT belső változóba
kaveautomata_label0:for (i = 0; i < N; i++)
                        T_VAL_MAT[i]=T_MATRIX[i];  //bemásoljuk az időzítésekhez rendelt értékeket

if (rst==1)
{
          allapot=RDY;
}

//while(end==0)
pipeline:{
switch(allapot) {

  case RDY :

          szamlalo=T_VAL_MAT[0];
          VEZERLO_VAL=KESZENLET;
            if (start==1)
                        allapot=P_LETESZ;
            else
                        allapot=RDY;
    break; /* optional */

  case P_LETESZ :
            VEZERLO_VAL=ELOKESZIT;
```

```
            szamlalo=T_VAL_MAT[0];
            allapot=V_MELEGIT;

    break; /* optional */


case V_MELEGIT :
            VEZERLO_VAL=VIZET_MELEGIT;
            szamlalo=szamlalo-1;
                        if (szamlalo==0)
                        { szamlalo=T_VAL_MAT[1];
                                    allapot=K_ADAGOL;}
                        else
                                    allapot=V_MELEGIT;
    break; /* optional */

case K_ADAGOL :
            VEZERLO_VAL=KAVET_ADAGOL;
            szamlalo=szamlalo-1;
                        if (szamlalo==0)
                                    {szamlalo=T_VAL_MAT[2];
                                    allapot=C_ADAGOL;}
                        else
                                    allapot=K_ADAGOL;
    break; /* optional */
case C_ADAGOL :
            VEZERLO_VAL=CUKROT_ADAGOL;
            szamlalo=szamlalo-1;
                        if (szamlalo==0)
                        { szamlalo=T_VAL_MAT[3];
                                    allapot=T_ADAGOL;}
                        else
                                    allapot=C_ADAGOL;
    break; /* optional */

case T_ADAGOL :
            VEZERLO_VAL=TEJET_ADAGOL;
            szamlalo=szamlalo-1;
                        if (szamlalo==0)
                        {  szamlalo=T_VAL_MAT[4];
                                    allapot=V_ADAGOL;}
                        else
                                    allapot=T_ADAGOL;
    break; /* optional */
case V_ADAGOL :
            VEZERLO_VAL=VIZET_ADAGOL;
            szamlalo=szamlalo-1;
                        if (szamlalo==0)
```

```
                    { szamlalo=T_VAL_MAT[5];
                            allapot=KEVER;}
                    else
                            allapot=V_ADAGOL;
    break; /* optional */

  case KEVER :
            VEZERLO_VAL=KEVERI;
                    szamlalo=szamlalo-1;
                    if (szamlalo==0)
                    { allapot=VARAKOZIK;}
                    else
                            allapot=KEVER;
  break; /* optional */

  case VARAKOZIK :
            VEZERLO_VAL=KESZ;

                    if (pf==1)
                            {szamlalo=T_VAL_MAT[1];
                            allapot=RDY;}
                    else
                            allapot=VARAKOZIK;
  break; /* optional */
}
//A beállított kimeneti értékeket értékeket kimásoljuk a kimeneti portjelekre

                    *VEZERLO_JELEK=VEZERLO_VAL;
//return VEZERLO_VAL;
return;
}
}
```

Header fájl

```
#ifndef _KAVE_AUTOMATA_H_
#define _KAVE_AUTOMATA_H_
#define N 7
//#include "systemc.h"
#include <stdio.h>

//typedef short din1_t;
typedef unsigned char din_t;
//typedef unsigned int dint_t;
typedef unsigned char dout_t;
typedef enum { RDY,P_LETESZ, V_MELEGIT, K_ADAGOL, C_ADAGOL, T_ADAGOL, V_ADAGOL, KEVER,
VARAKOZIK} allapot_tipus;
```

```
void kaveautomata(din_t T_MATRIX[6], dout_t  *VEZERLO_JELEK, din_t rst, din_t start, din_t pf);
//void szemafor(din_t T_MATRIX[4], dout_t  SZEMAFOR_OUT[4]);

#endif
```

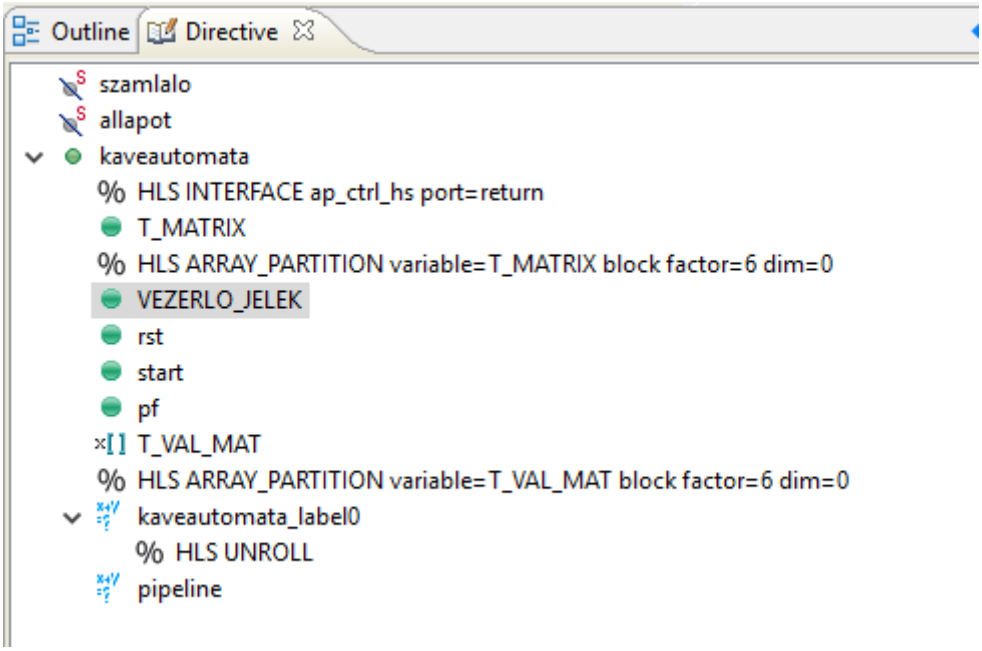
TestBench állomány a modulnak szoftver szinten való teszteléséhez

```
#include "kave_automata.h"
int main()
{
	din_t T_MATRIX[N]={10,4,2,2,8,10};
	din_t *T_MATRIX_POINTER;
	din_t start;
	din_t reset;
	din_t end;
	dout_t  VEZERLO_JELEK;
	T_MATRIX_POINTER=T_MATRIX;
	int i, retval=0;
		FILE	*fp;
		// Save the results to a file
		fp=fopen("result1.dat","w");

		// Call the function for multiple transactions
		for (i=0; i<200; i++)
		{
		switch (i)
		{
			case 0:  kaveautomata(T_MATRIX,&VEZERLO_JELEK, 1, 0, 0);
					break;
			case 1:  kaveautomata(T_MATRIX, &VEZERLO_JELEK, 0, 1, 0);
					break;
			case 50:	kaveautomata(T_MATRIX, &VEZERLO_JELEK, 0, 1, 1);
			//case 100:	kaveautomata(T_MATRIX, &VEZERLO_JELEK, 0, 1, 1);
								break;
			default:	kaveautomata(T_MATRIX,&VEZERLO_JELEK, 0, 1, 0);
		}
		fprintf(fp, "%x\n", VEZERLO_JELEK);
		}
		fclose(fp);
return 0;
}
```

Alkalmazott direktívák:

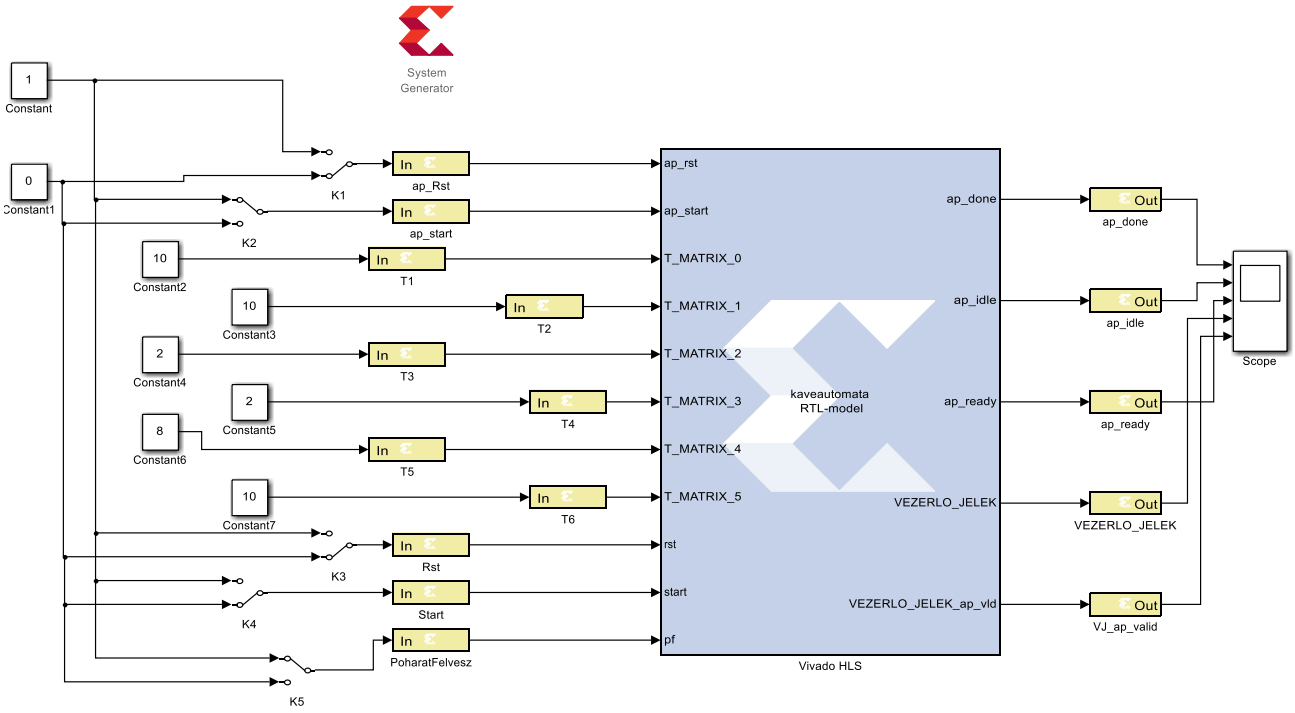
set_directive_interface -mode ap_ctrl_hs "kaveautomata"
set_directive_array_partition -type block -factor 6 -dim 0 "kaveautomata" T_MATRIX
set_directive_unroll "kaveautomata/kaveautomata_label0"
set_directive_array_partition -type block -factor 6 -dim 0 "kaveautomata" T_VAL_MAT



A fentebb felsorolt direktívákkal a generált IP mag a következő jeleket tartalmazza:

RTL Ports	Dir	Bits	Protocol	Source Object	C Type
ap_clk	in	1	ap_ctrl_hs	kaveautomata	return value
ap_rst	in	1	ap_ctrl_hs	kaveautomata	return value
ap_start	in	1	ap_ctrl_hs	kaveautomata	return value
ap_done	out	1	ap_ctrl_hs	kaveautomata	return value
ap_idle	out	1	ap_ctrl_hs	kaveautomata	return value
ap_ready	out	1	ap_ctrl_hs	kaveautomata	return value
T_MATRIX_0	in	8	ap_none	T_MATRIX_0	pointer
T_MATRIX_1	in	8	ap_none	T_MATRIX_1	pointer
T_MATRIX_2	in	8	ap_none	T_MATRIX_2	pointer
T_MATRIX_3	in	8	ap_none	T_MATRIX_3	pointer
T_MATRIX_4	in	8	ap_none	T_MATRIX_4	pointer
T_MATRIX_5	in	8	ap_none	T_MATRIX_5	pointer
VEZERLO_JELEK	out	8	ap_vld	VEZERLO_JELEK	pointer
VEZERLO_JELEK_ap_vld	out	1	ap_vld	VEZERLO_JELEK	pointer
rst	in	8	ap_none	rst	scalar
start	in	8	ap_none	start	scalar
pf	in	8	ap_none	pf	scalar

A System Generator modul, amely alapján le lehet futtatni az IP mag szimulációját:



Szimuláció eredménye