TUDELFT
Faculty Electrical Engineering, Mathematics and Computer Science
MSc Embedded Systems 2011-2013

ET 4054 Methods and Algorithms for System Design

Lab Report

Name: Charalampos Papadopoulos
St ID: 4192141

Introduction

This laboratory exercise illustrates the design of hardware starting from a 'high level' description of a circuit. We start with translating a number of circuits in Data Flow Graphs, which consists of operations, delays and interconnect. Using MATLAB, we schedule (using various scheduling methods, resource constraints, resource latencies, etc) the circuit descriptions. Again, using MATLAB, it is possible to test a circuit whether 'retiming for minimum clock states' can result in 'faster' circuits. For each circuit given, Pareto points are obtained and plotted. After that we select a 'best' circuit that we use for further investigation. The 'best' function in this context can be the smallest, fastest, or even the most simple implementation.

Still using MATLAB, we simulate the scheduled circuit and calculate output values depending on user-definable input sequences (these values can be used later on to verify that the 'hardware'-simulation operates correctly). Another MATLAB program generates corresponding bit-true VHDL code for the data path and the controller part of your circuit, together with a so-called testbench. This testbench and circuit code will have to be simulated with the aid of the VHDL simulator Model*Sim*. Evidently, the outputs of both MATLAB and the VHDL testbench should be in accordance with each other.

Selected circuit list

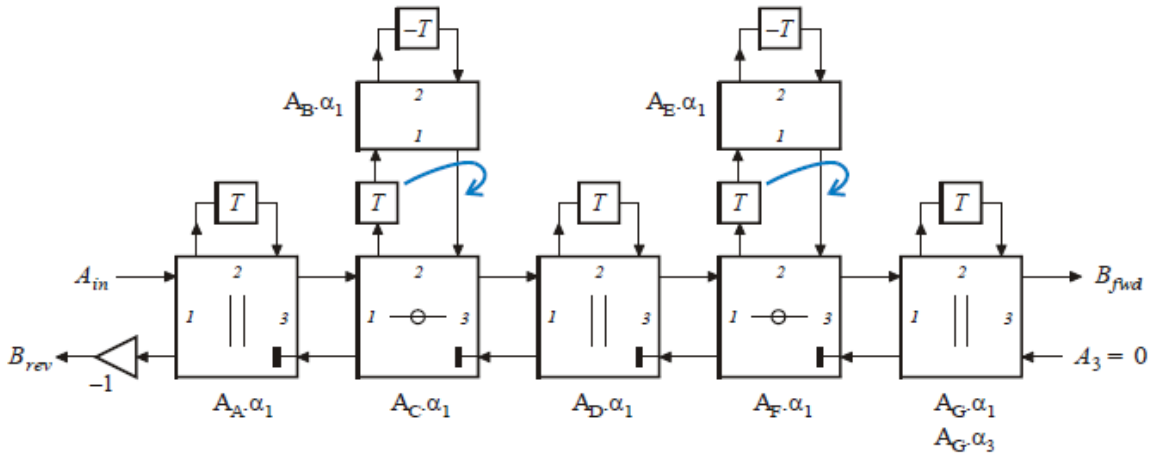
The circuits that are going to be studied are the following.

WDF_C5_p_2p_L (see Figure 2.6a)
WDF_C5_p_2p_R (see Figure 2.6a)
WDF_C5_p_3p (see Figure 2.6b)
WDF_C5_s_sym_2p_L (see Figure 2.7a)
WDF_C5_s_sym_2p_R (see Figure 2.7a)
LWDF_5_L (see Figure 2.9a)
LWDF_5_R (see Figure 2.9a)
LWDF_5_T_L (see Figure 2.9b)
LWDF_5_T_R (see Figure 2.9b)

Here we should notice that the circuits from Figure 2.9b are the pipelined version of the circuits of Figure 2.9a.

Circuit_WDF_C5_p_2p_L

This circuit belongs to the image below, when the two registers are at the left side.

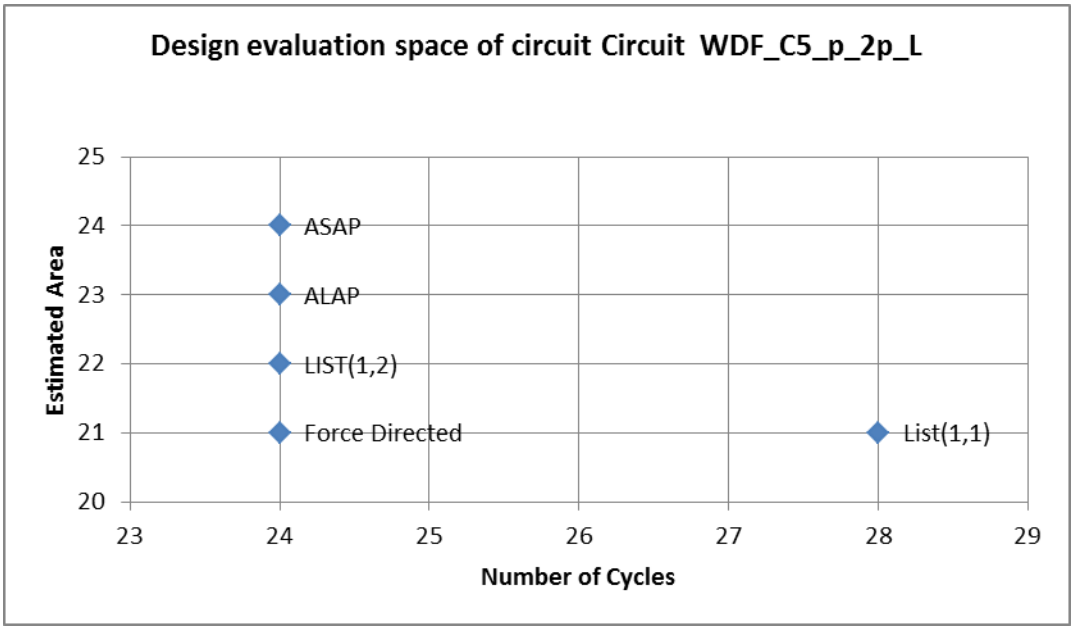


$\alpha_1 A$	0.61567453576615
$\alpha_1 B$	0.11760445944132
$\alpha_1 C$	0.46968595194661
$\alpha_1 D$	0.43378869238454
$\alpha_1 E$	0.24794776416762
$\alpha_1 F$	0.52818356937584
$\alpha_1 G$	0.76350613851909
$\alpha_3 G$	0.82195774820984

Figure 2.6a 'WDF_C5_p_2p_L' and 'WDF_C5_p_2p_R'

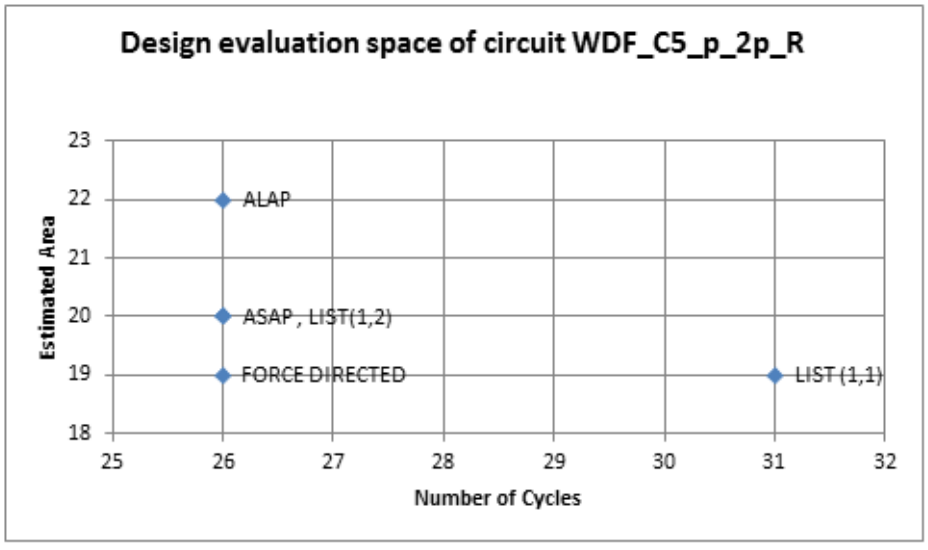
We investigate the scheduling properties of the above circuit using the schedGUI program.

Sched Method	# Mul	#ALUs	#Registers	Total Area	Cycles
ASAP	3	3	18	24	24
ALAP	1	4	18	23	24
Force Directed	1	2	18	21	24
List(1 M,1 A)	1	1	19	21	28
List(1M ,2 A)	1	2	19	22	24



Circuit WDF_C5_p_2p_R

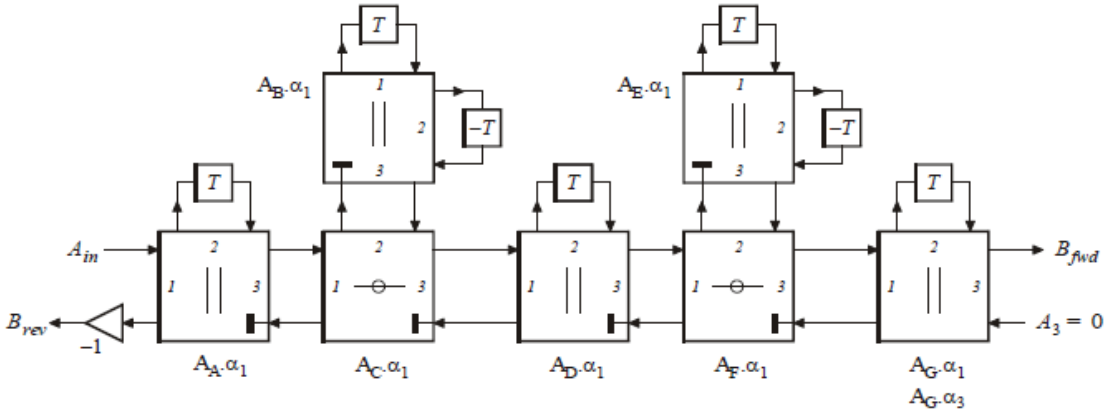
Sched Method	# Mul	#ALUs	#Registers	Total Area	Cycles
ASAP	1	2	17	20	26
ALAP	2	4	16	22	26
Force Directed	1	2	16	19	26
List(1 M,1 A)	1	1	17	19	31
List(1M ,2 A)	1	2	17	20	26



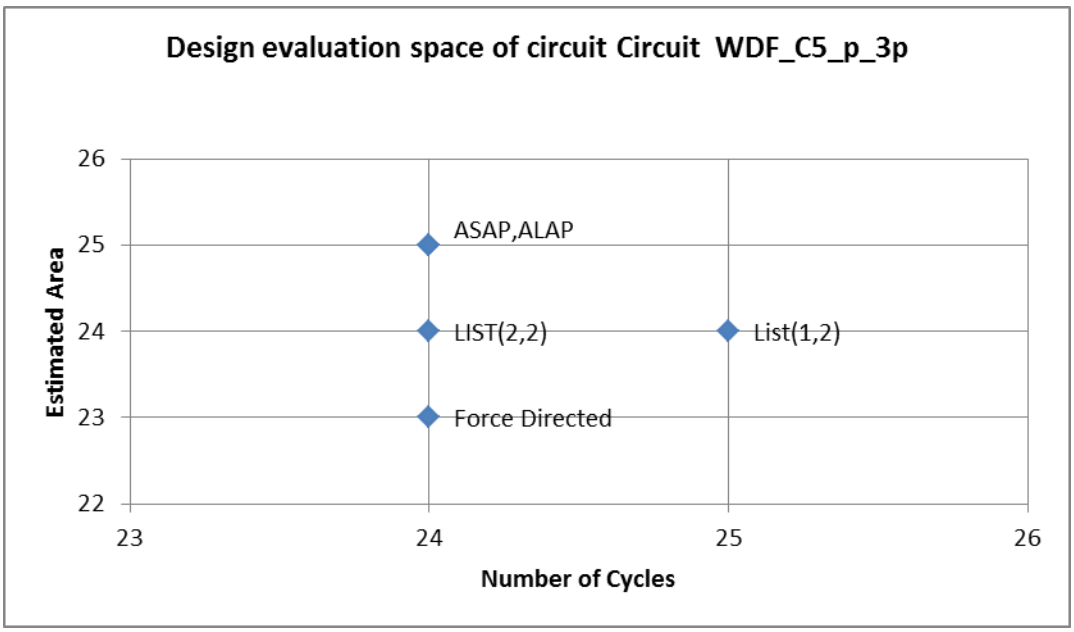
Circuit_WDF_C5_p_3p

$\alpha_1 A$	0.61567453576615
$\alpha_1 B$	0.05880222972066
$\alpha_1 C$	0.46968595194661
$\alpha_1 D$	0.43378869238454
$\alpha_1 E$	0.12397388208381
$\alpha_1 F$	0.52818356937584
$\alpha_1 G$	0.76350613851909
$\alpha_3 G$	0.82195774820984

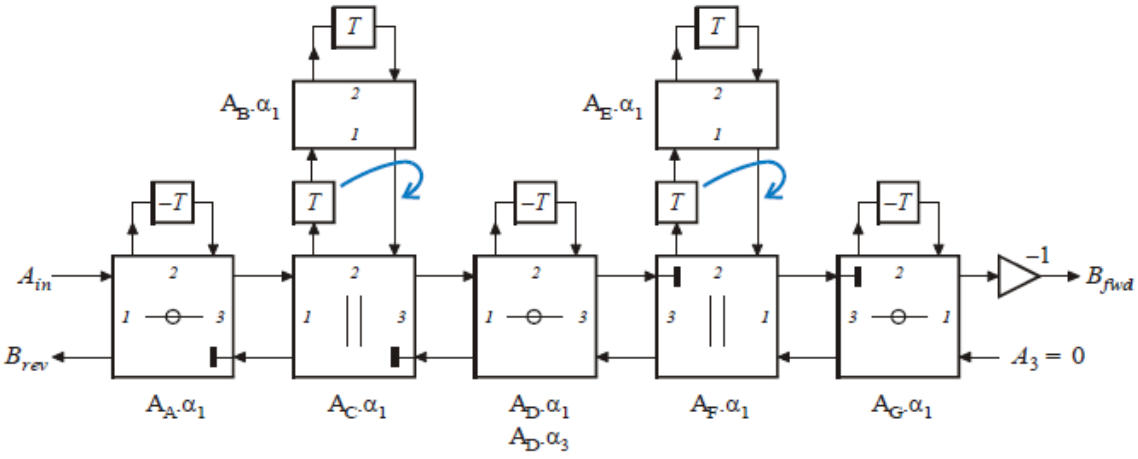
Figure 2.6b 'WDF_C5_p_3p'



Sched Method	# Mul	#ALUs	#Registers	Total Area	Cycles
ASAP	3	3	19	25	24
ALAP	2	4	19	25	24
Force Directed	2	2	19	23	24
List(1 M,2 A)	1	2	21	24	25
List(2M ,2 A)	2	2	20	24	24



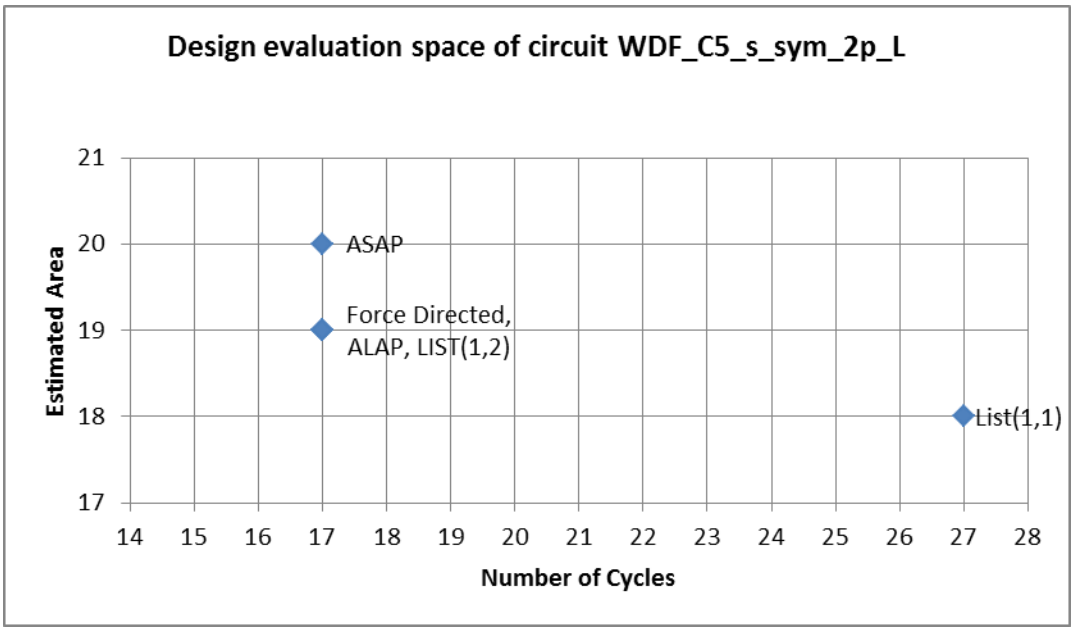
Circuit WDF_C5_s_sym_2p_L



$\alpha_1 A$	0.61567453576615
$\alpha_1 B$	1.88239554055869
$\alpha_1 C$	0.46968595194661
$\alpha_1 D$	0.58427183716240
$\alpha_3 D$	0.65309574126830
$\alpha_1 E$	1.75205223583238
$\alpha_1 F$	0.56685999288327
$\alpha_1 G$	0.66474874952101

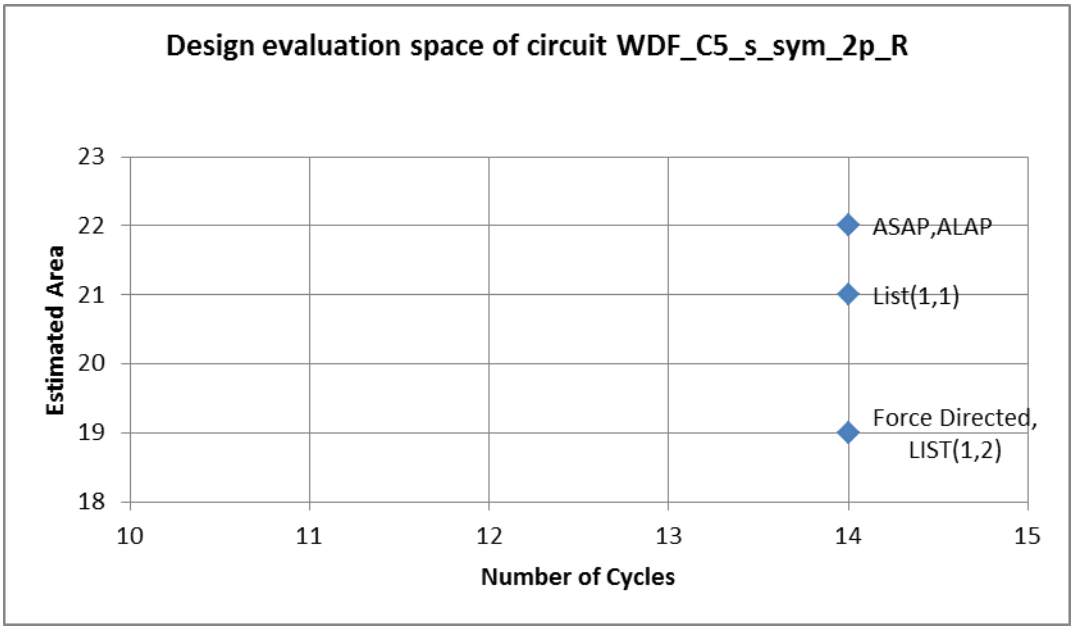
Figure 2.7a 'WDF_C5_s_sym_2p_L' and 'WDF_C5_s_sym_2p_R'

Sched Method	# Mul	#ALUs	#Registers	Total Area	Cycles
ASAP	2	3	15	20	17
ALAP	1	3	15	19	17
Force Directed	1	2	16	19	17
List(1 M,1 A)	1	1	16	18	27
List(1M ,2 A)	1	2	16	19	17



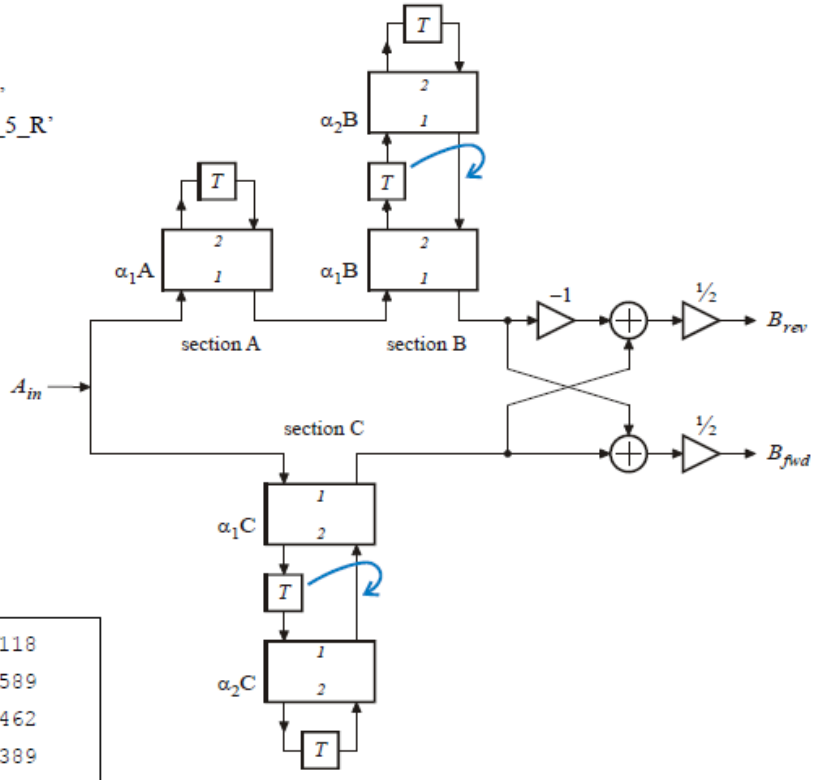
Circuit WDF_C5_s_sym_2p_R

Sched Method	# Mul	#ALUs	#Registers	Total Area	Cycles
ASAP	2	5	15	22	14
ALAP	3	5	14	22	14
Force Directed	2	4	13	19	14
List(2 M,4 A)	2	4	15	21	14
List(2M ,3 A)	2	3	14	19	14



Circuit_LWDF_5_L

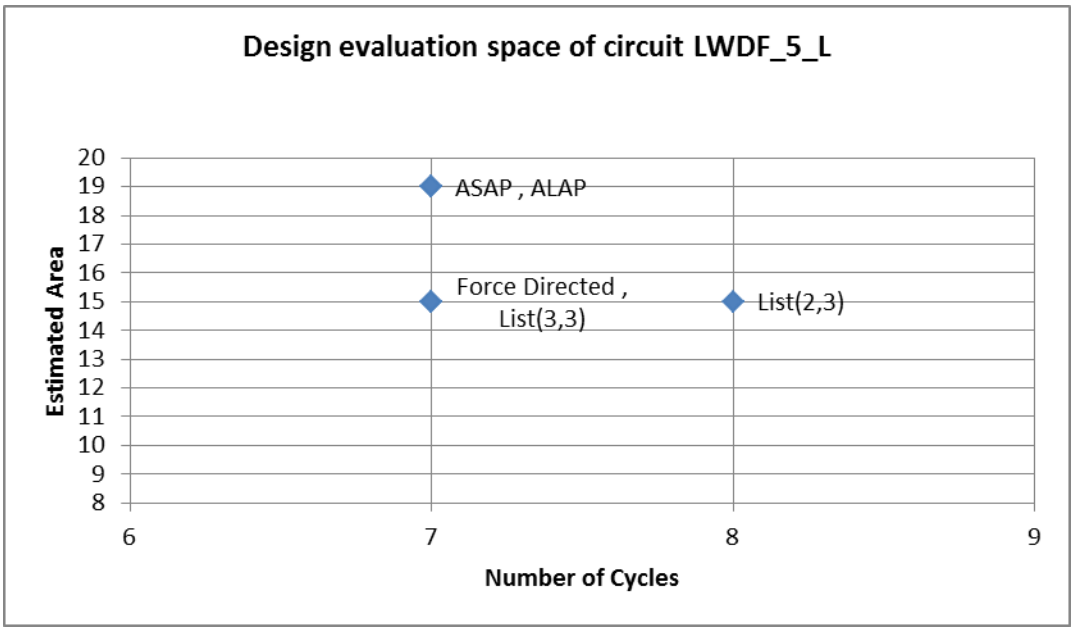
Figure 2.9a 'LWDF_5_L' and 'LWDF_5_R'



Coefficients for both 'LWDF_5_L|R' *and* 'LWDF_5_T_L|R'.

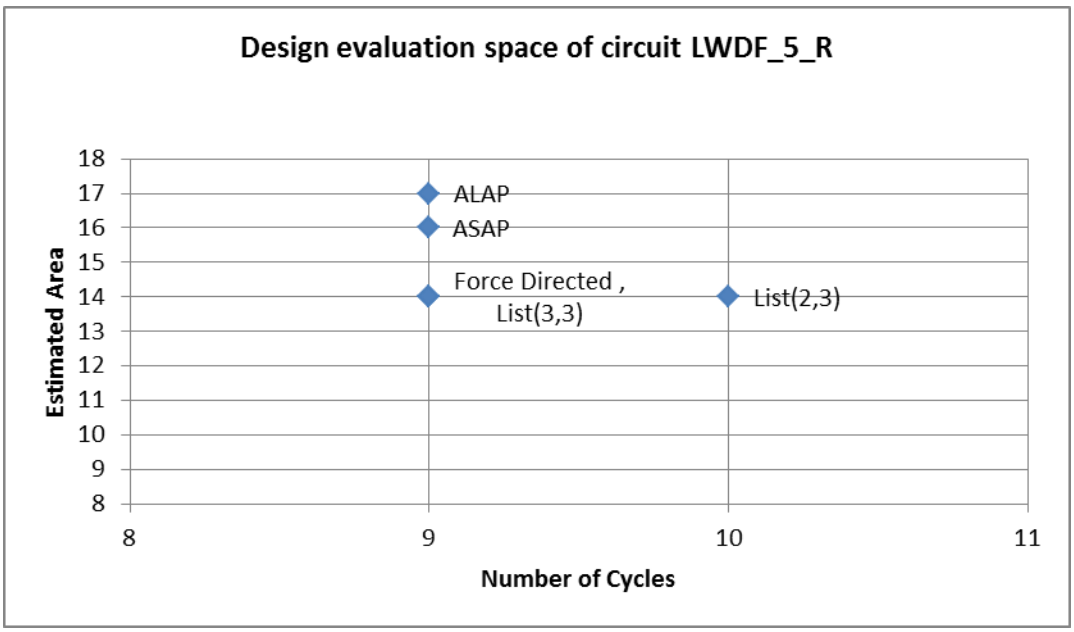
$\alpha_1 A$	-0.03119210537118
$\alpha_1 B$	-0.82935592147589
$\alpha_2 B$	-0.52095844720462
$\alpha_1 C$	-0.38195386087389
$\alpha_2 C$	-0.35687921966298

Sched Method	# Mul	#ALUs	#Registers	Total Area	Cycles
ASAP	3	6	10	19	7
ALAP	3	7	9	19	7
Force Directed	3	3	9	15	7
List(3 M,3 A)	3	3	9	15	7
List(2M ,3 A)	2	3	10	15	8



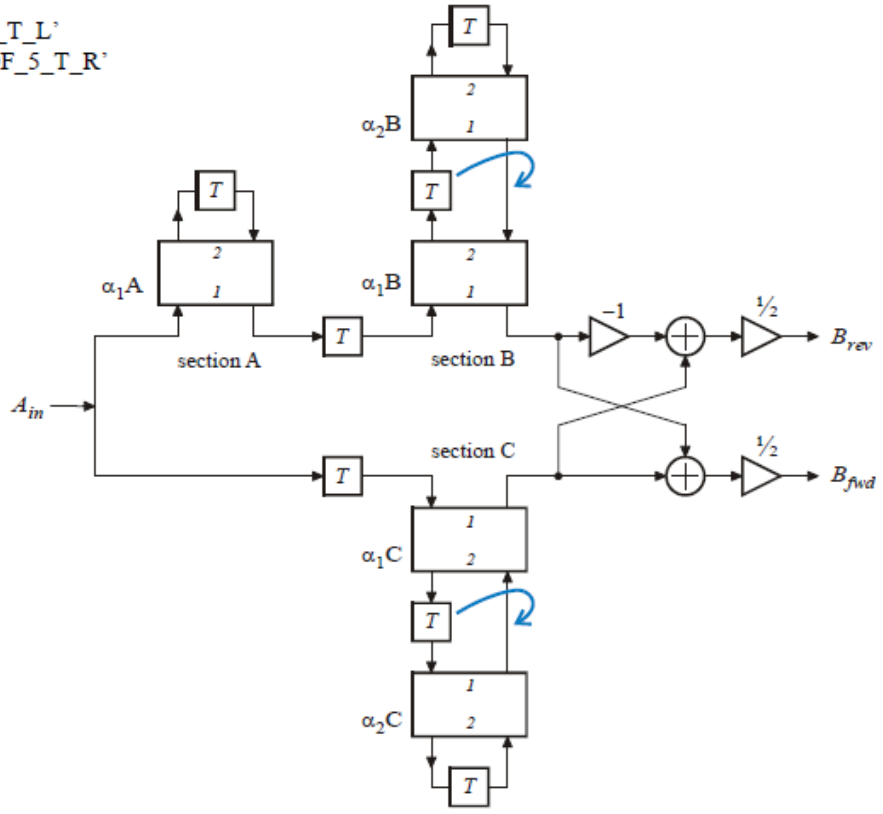
Circuit_LWDF_5_R

Sched Method	# Mul	#ALUs	#Registers	Total Area	Cycles
ASAP	2	4	10	16	9
ALAP	2	7	8	17	9
Force Directed	1	3	10	14	9
List(1 M,3 A)	1	3	10	14	9
List(1M ,2 A)	1	2	11	14	10

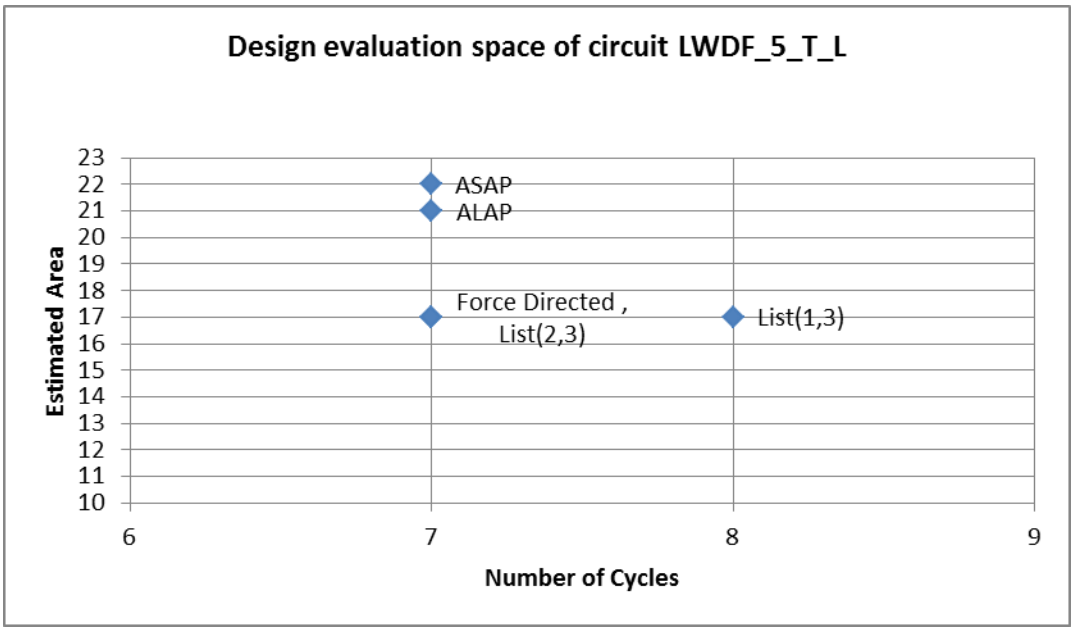


Circuit_LWDF_5_T_L

Figure 2.9b 'LWDF_5_T_L' and 'LWDF_5_T_R'

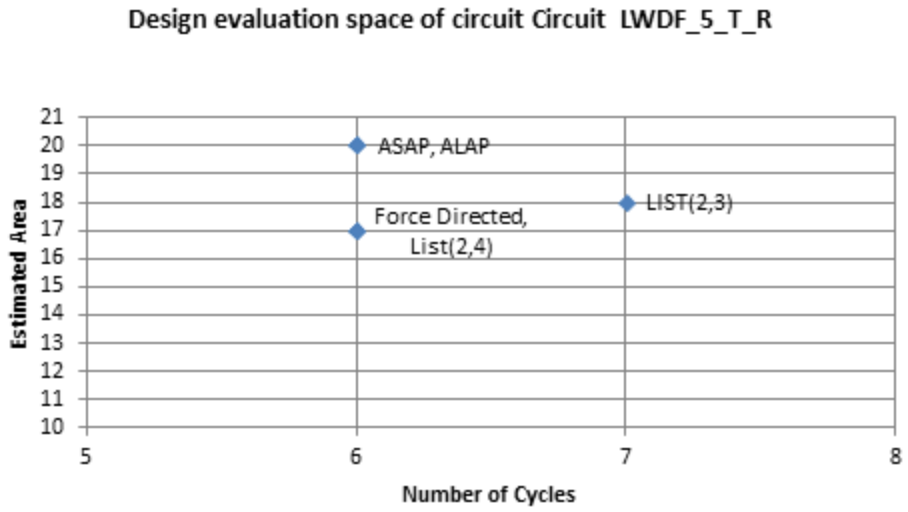


Sched Method	# Mul	#ALUs	#Registers	Total Area	Cycles
ASAP	3	6	13	22	7
ALAP	2	8	11	21	7
Force Directed	2	3	12	17	7
List(2 M,3 A)	2	3	12	17	7
List(1M ,3 A)	1	3	13	17	8



Circuit LWDF_5_T_R

Sched Method	# Mul	#ALUs	#Registers	Total Area	Cycles
ASAP	3	6	11	20	6
ALAP	3	8	9	20	6
Force Directed	2	4	11	17	6
List(2 M,4 A)	2	4	11	17	6
List(2M ,3 A)	2	3	13	18	7



Retiming of circuits

Below we investigate which of the above circuits can be retimed and which not. We use the matlab function retime_minCycles developed by TUDelft. The results are shown below:

WDF_C5_p_2p_L

Originally, the longest path needs 24 states

Retiming possible to decrease longest path to 23 state(s)

WDF_C5_p_2p_R

Originally, the longest path needs 26 states

Retiming possible to decrease longest path to 23 state(s)

WDF_C5_p_3p

Originally, the longest path needs 24 states

Sorry, no retiming possible at all ...

WDF_C5_s_sym_2p_L

Originally, the longest path needs 17 states

Retiming possible to decrease longest path to 14 state(s)

Decreasing longest path to 13 cycles is not feasible ...

WDF_C5_s_sym_2p_R

Originally, the longest path needs 14 states

Sorry, no retiming possible at all ...

LWDF_5_L

Originally, the longest path needs 7 states

Retiming possible to decrease longest path to 6 state(s)

Decreasing longest path to 5 cycles is not feasible ...

LWDF_5_R

Originally, the longest path needs 9 states

Retiming possible to decrease longest path to 6 state(s)

Decreasing longest path to 5 cycles is not feasible ...

LWDF_5_T_L

Originally, the longest path needs 7 states

Retiming possible to decrease longest path to 6 state(s)

Decreasing longest path to 5 cycles is not feasible ...

LWDF_5_T_R

Originally, the longest path needs 6 states

Sorry, no retiming possible at all ...

Conclusions from the above

- Best scheduling algorithm

The force directed scheduling algorithm is always the best one out of the algorithms we tried, in terms of area needed and cycles achieved. The second best algorithm is list scheduling.

- a_L or a_R implementation is better

Below we provide a table comparison of the 4 circuit pairs. For the comparison we use the results obtained by the directed scheduling algorithm for all circuits.

Pair	L	R	Better
1	Area 21	Area 19	R
	Cycles 24	Cycles 26	L
2	Area 19	Area 19	=
	Cycles 17	Cycles 14	R
3	Area 15	Area 14	R
	Cycles 7	Cycles 9	L
4	Area 17	Area 17	=
	Cycles 7	Cycles 6	R

Obviously, we cannot say that one of the two implementations is better. Furthermore, after retiming of the two versions, they both have the same amount of cycles.

Matlab Testbench

Still using MATLAB, we simulate the scheduled circuit and calculate output values depending on user-definable input sequences (these values can be used later on to verify that your 'hardware'-simulation operates correctly).
Another MATLAB program generates corresponding bit-true VHDL code for the data path and the controller part of the circuit, together with a so-called testbench. This testbench and circuit code are simulated with the aid of Model*Sim*. Evidently, the outputs of both MATLAB and the VHDL testbench should be in accordance with each other. Here, we don't impose accuracy constraints and simply opt for a representation with 15 fractional bits.

The circuit that we are going to simulate is circuit LWDF_5_R , as it is the one which seems to be the simplest one.

Results from matlab testbench:

```
MATLAB testbench output: 15-Mar-2013 17:14:06
SFxd3 = [17 15 0]
Outputs resp.: o_hp, o_lp
x"00000"
x"00000"
---
x"016CA"
x"01A19"
---
x"1F4F2"
x"062F5"
---
x"018EE"
x"1990E"
---
x"1F20E"
x"18499"
---
x"001B5"
x"06FCD"
---
x"00484"
x"18632"
---
x"1F980"
x"186BE"
---
x"004E6"
x"076E1"
---
x"1FFCD"
```

```
x"1822C"
---
x"1FBA2"
x"184FE"
---
```

The input to the tested circuit is shown below.

```
-- [17 15] fixed-point format
-- coefficients: a1A,a1B,a1C,a2B,a2C
x"1FC02"
x"195D8"
x"1CF1C"
x"1BD51"
x"1D252"
-- input function
x"08000"
x"08000"
x"08000"
x"08000"
x"08000"
x"08000"
x"08000"
x"08000"
x"08000"
x"08000"
```

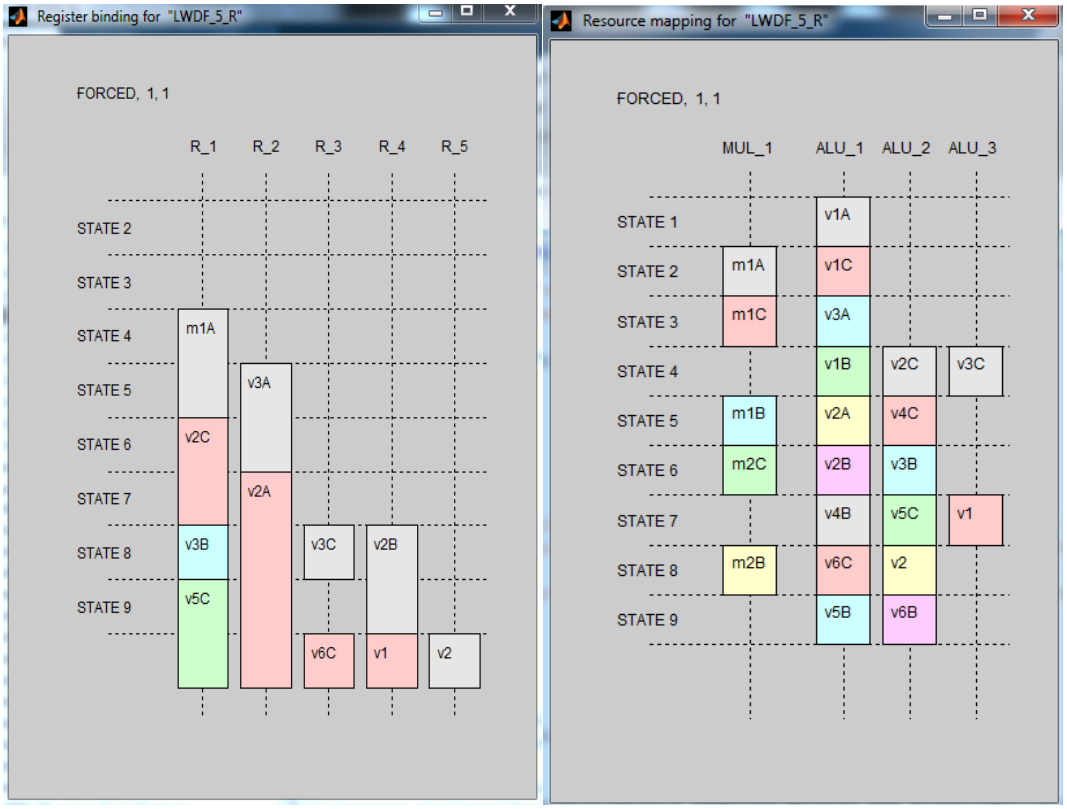
The output of the matlab tested circuit is shown below.

```
MATLAB testbench output: 15-Mar-2013 17:14:06
SFxd3 = [17 15 0]
Outputs resp.: o_hp, o_lp
x"00000"
x"00000"
---
x"016CA"
x"01A19"
---
x"1F4F2"
x"062F5"
---
x"018EE"
x"1990E"
---
x"1F20E"
x"18499"
---
x"001B5"
x"06FCD"
```

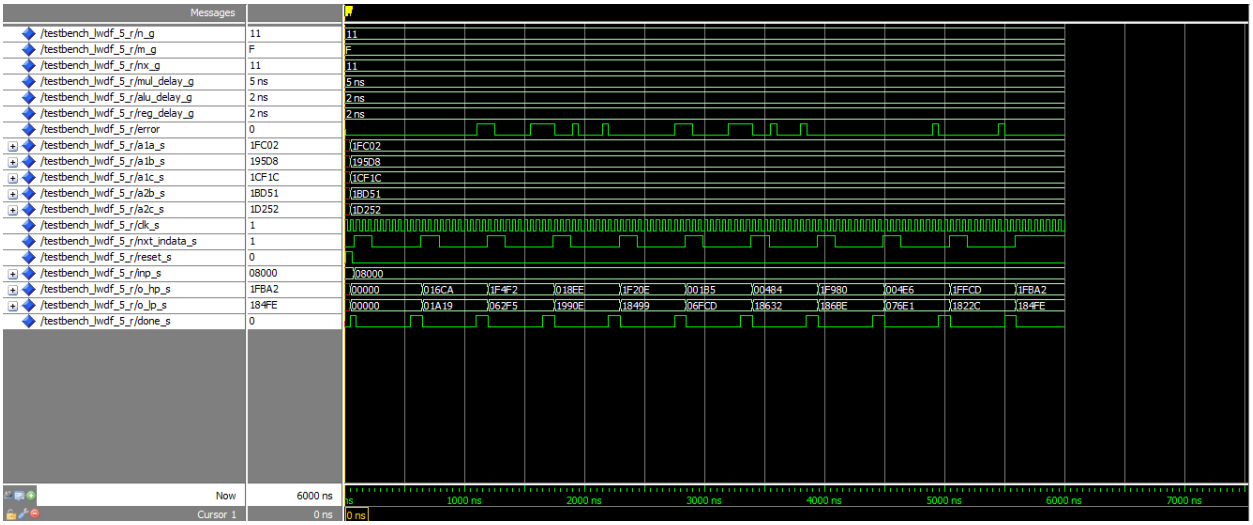
```
---
x"00484"
x"18632"
---
x"1F980"
x"186BE"
---
x"004E6"
x"076E1"
---
x"1FFCD"
x"1822C"
---
x"1FBA2"
x"184FE"
---
```

VHDL Testbench

Below we can see the results of the Vhdl test bench automatically generated by Matlab for the above circuit:



The output of the circuit can be seen below:



We can observe that the output of the vhdl testbench is the same with that of the Matlab testbench.

Conclusions

This lab provided an excellent tool in order to experiment on digital circuits concerning scheduling algorithms and retiming. The code which has been developed at TUDelft can be used to experiment with digital circuits in order to take optimal implementation decisions. The validity of the generated code has been verified as described above.