



Układy sekwencyjne

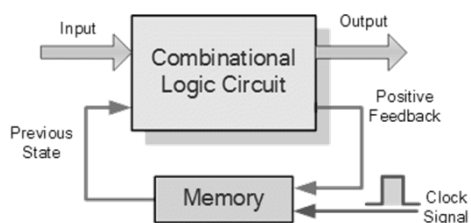
Technika cyfrowa i mikroprocesorowa
dr inż. Dominik Łuczak

1



Poznan University of Technology
Institute of Control, Robotics and Information Engineering

Sequential logic



2019-10-12

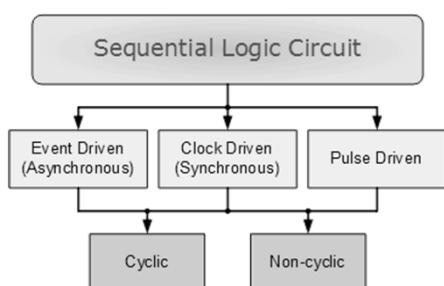
dominik.luczak@pwr.edu.pl

2

2



Poznan University of Technology
Institute of Control, Robotics and Information Engineering

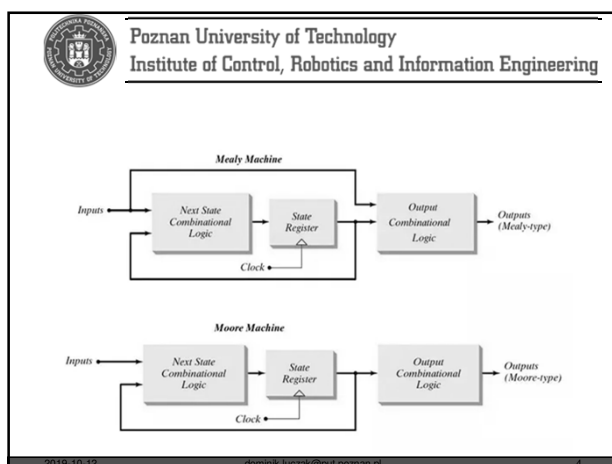


2019-10-12

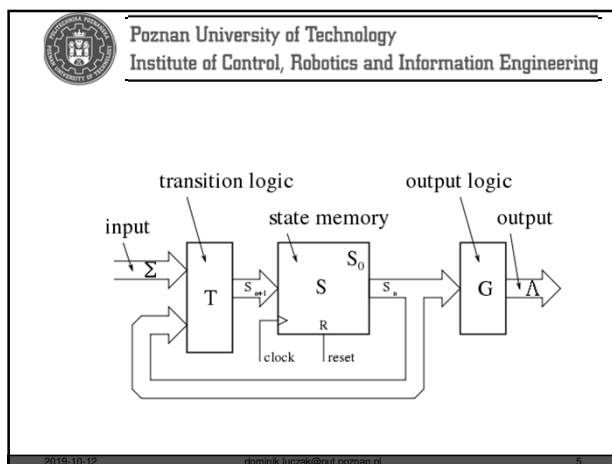
dominik.luczak@pwr.edu.pl

3

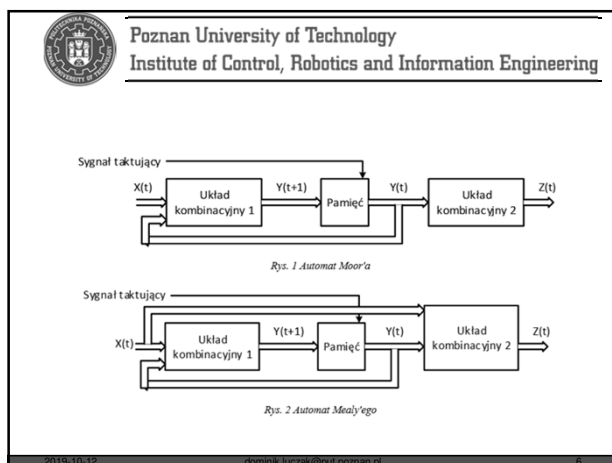
3



4



5



6



State memory

2019-10-12

dominik.luczak@put.poznan.pl

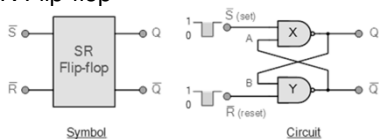
7

7



Poznan University of Technology
Institute of Control, Robotics and Information Engineering

Basic SR Flip-flop



State	S	R	Q	Description
Set	1	0	0	Set $Q = 1$
	1	1	0	no change
Reset	0	1	1	Reset $Q = 0$
	1	1	1	no change
Invalid	0	0	1	Invalid Condition

2019-10-12

dominik.luczak@put.poznan.pl

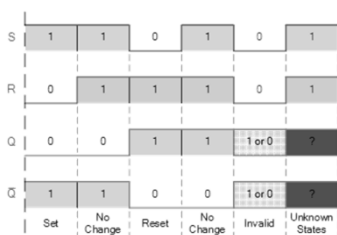
8

8



Poznan University of Technology
Institute of Control, Robotics and Information Engineering

S-R Flip-flop Switching Diagram



2019-10-12

dominik.luczak@put.poznan.pl

9

9

Poznan University of Technology
Institute of Control, Robotics and Information Engineering

The NOR Gate SR Flip-flop

S	R	Q	$\bar{Q}$
0	0	No change	
0	1	1	0
1	0	0	1
1	1	X	X
		(Invalid)	

NOR Circuit

2019-10-12 dominik.kucinski@pwr.edu.pl 10

10

Poznan University of Technology
Institute of Control, Robotics and Information Engineering

Active HIGH Flip-flops

S	R	Q	$\bar{Q}$
0	0	No change	
0	1	0	1
1	0	1	0
1	1	X	X
		(Invalid)	

NAND Circuit

2019-10-12 dominik.kucinski@pwr.edu.pl 11

11

Poznan University of Technology
Institute of Control, Robotics and Information Engineering

2019-10-12 dominik.kucinski@pwr.edu.pl 12

12

Poznan University of Technology
Institute of Control, Robotics and Information Engineering

SR Flip Flop Switch Debounce Circuit

2019-10-12 domini.k.luczak@pwr.edu.pl 13

13

Poznan University of Technology
Institute of Control, Robotics and Information Engineering

Debounce Circuit

RC Switch Debounce Circuit Switch Debounce with NAND Gates

2019-10-12 domini.k.luczak@pwr.edu.pl 14

14

Poznan University of Technology
Institute of Control, Robotics and Information Engineering

Gated SR Flip-flop Circuit

Tab. 3 Tabela przejść przerzutnika typu RS

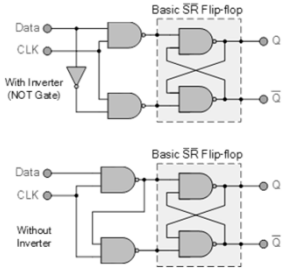
SR	$Q \rightarrow Q^{n+1}$
00	$Q \rightarrow Q^{n+1}$
10	$Q \rightarrow 1$
01	$Q \rightarrow 0$
11	$Q \rightarrow -$

2019-10-12 domini.k.luczak@pwr.edu.pl 15

15

 Poznan University of Technology
Institute of Control, Robotics and Information Engineering

The D-type Flip-flop (NAND)

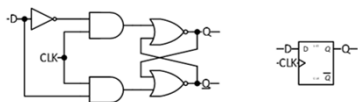


2019-10-12 domini.k.kucinski@pwr.lodz.pl 16

16

 Poznan University of Technology
Institute of Control, Robotics and Information Engineering

D (AND, NOR)



Rys. 4 Przerzutnik typu D

Tab. 4 Tabela przejść przerzutnika typu D

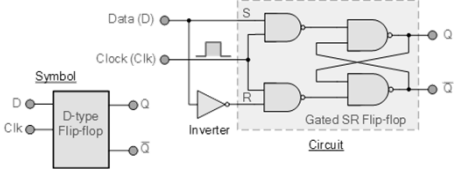
D	$Q^n \rightarrow Q^{n+1}$
0	$0 \rightarrow 0$
0	$1 \rightarrow 0$
1	$0 \rightarrow 1$
1	$1 \rightarrow 1$

2019-10-12 domini.k.kucinski@pwr.lodz.pl 17

17

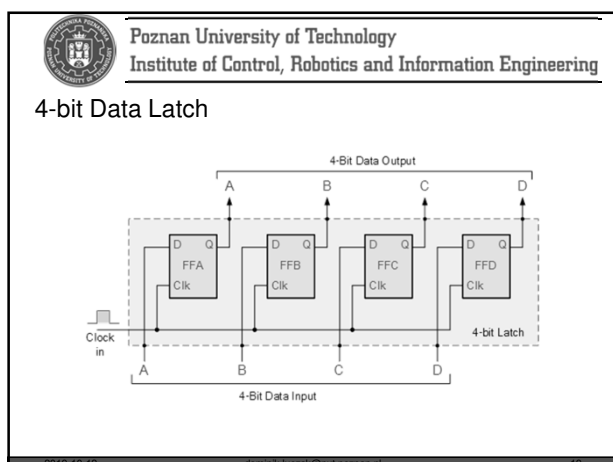
 Poznan University of Technology
Institute of Control, Robotics and Information Engineering

D-type Flip-flop

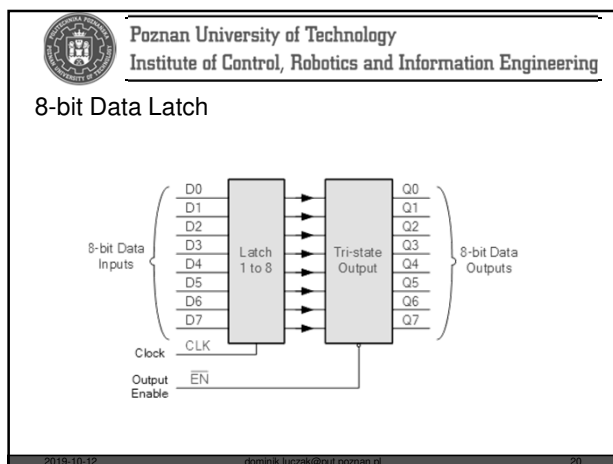


2019-10-12 domini.k.kucinski@pwr.lodz.pl 18

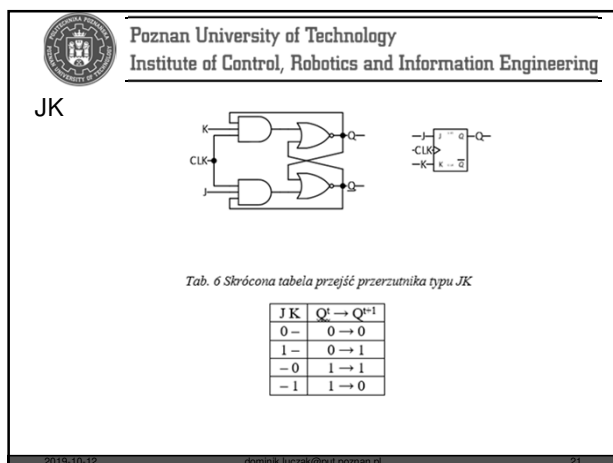
18



19



20



21



Przykład

2019-10-12

dominik.luczak@put.poznan.pl

22

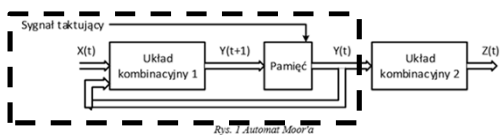
22



Poznan University of Technology
Institute of Control, Robotics and Information Engineering

Tab. 2 Tabela przejść automatu

S	$\sigma=0$	$\sigma=1$
0	0	3
1	4	3
2	2	5
3	0	4
4	1	2
5	1	5



2019-10-12

dominik.luczak@put.poznan.pl

23

23



Poznan University of Technology
Institute of Control, Robotics and Information Engineering

Tab. 7 Realizowany automat

S	$\sigma=0$	$\sigma=1$	λ
1	1	2	0
2	2	3	1
3	3	2	1

Tab. 8 Kodowanie stanów automatu

S	$s_1 s_2$
0	00
1	01
2	10
3	11

Tab. 9 Realizowany automat z kodowaniem stanów

$s_1 s_2$	$\sigma=0$	$\sigma=1$	λ
00	-	-	-
01	01	10	0
10	10	11	1
11	11	10	1

Tab. 10 Zapis w pamięci stałej

Nr	A2	A1	A0	Q2	Q1	Q0
σ	s_1'	s_2'	s_1^{t+1}	s_2^{t+1}	λ	
0	0	0	0	-	-	-
1	0	0	1	0	1	0
2	0	1	0	1	0	1
3	0	1	1	1	1	1
4	1	0	0	-	-	-
5	1	0	1	1	0	0
6	1	1	0	1	1	1
7	1	1	1	1	0	1

2019-10-12

dominik.luczak@put.poznan.pl

24

24

Poznan University of Technology
Institute of Control, Robotics and Information Engineering

2019-10-12 domini.kuczyński@pwr.edu.pl 25

25

Poznan University of Technology
Institute of Control, Robotics and Information Engineering

$$s_1^{n+1} = D_1 = s_1 + \sigma = s_1 + \overline{s_2} \cdot \overline{\sigma} \quad (2)$$

$$s_2^{n+1} = D_2 = s_2 \cdot \overline{\sigma} + \overline{s_2} \cdot \sigma = s_2 \cdot \overline{\sigma} + \overline{s_2} \cdot \sigma = s_2 \cdot \overline{\sigma} + \overline{s_2} \cdot \sigma \quad (3)$$

$$\lambda = s_1 \quad (4)$$

Tab. 11 Realizowany automat z kodowaniem stanów - tablica Karnaugh

s_1	s_2	$\sigma=0$	$\sigma=1$	λ
0	0	-	-	-
0	1	0	1	0
1	1	1	1	1
1	0	1	0	1
		D_1	D_2	D_1

2019-10-12 domini.kuczyński@pwr.edu.pl 26

26

Poznan University of Technology
Institute of Control, Robotics and Information Engineering

Tab. 12 Pomocnicza tablica Karnaugh - D1

s_1	s_2	$\sigma=0$	$\sigma=1$	λ
0	0	-	-	-
0	1	0	1	0
1	1	1	1	1
1	0	1	0	1
		D_1	D_2	

Tablica pomocnicza D1

Pierwszy fragment

s_1	s_2	$\sigma=0$	$\sigma=1$	λ
0	0	-	-	-
0	1	0	1	0
1	1	1	1	1
1	0	1	0	1
		D_1	D_2	

Drugi fragment

Tab. 13 Pomocnicza tablica Karnaugh - D2

s_1	s_2	$\sigma=0$	$\sigma=1$	λ
0	0	-	-	-
0	1	1	0	0
1	1	1	0	1
1	0	0	1	1
		D_2	D_2	

Tablica pomocnicza D2

Pierwszy fragment

Drugi fragment

2019-10-12 domini.kuczyński@pwr.edu.pl 27

27

28

The screenshot shows a Windows desktop with a Visual Studio Code editor window titled "https://www.online-gdb.com/online_c_compiler". The editor contains a C program that calculates the sum of squares of integers from 1 to 10. The program is as follows:

```

1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int* t1[] = {0,1,2,3,4,5,6,7,8,9};
5 int* t2[] = {0,1,2,3,4,5,6,7,8,9};
6
7 int* t1_0_52 = <signame>; //address=0, y=0.52, mem=0;
8
9 int main()
10 {
11     //Combination
12     for(<signame>=<signame>)
13         for(<S1>=0; <S1>=1; <S1>++)
14             for(<S2>=0; <S2>=1; <S2>++)
15                 {
16                     address = <signame> + <S1> * 1 + <S2>;
17                     S1=<address>;
18                     y=t[address];
19                     //C10k0, y=t[0];t[0]=t[<signame>+0]; //y,y,t[0]=<signame>+0,<S2>;
20                     mem++;
21                 }
22 //Sumup
23 S1=<signame>+1;
24 for(<int> i=0; i=<signame>+1; i++)
25     {
26         S1=<signame>+1;
27         S1=<address>;
28         y=t[address];
29         //C10k0, y=t[0];t[0]=t[<signame>+0]; //y,y,t[0]=<signame>+0,<S2>;
30         i(<int>)=<signame>;
31     }
32     return 0;
33 }

```

The output window on the right shows the execution of the program. It displays the values of the variables and the final result of the calculation. The output is as follows:

```

0.52 mem=0
0.52 mem=1
0.52 mem=2
0.52 mem=3
0.52 mem=4
0.52 mem=5
0.52 mem=6
0.52 mem=7
0.52 mem=8
0.52 mem=9
0.52 mem=10
0.52 mem=11
0.52 mem=12
0.52 mem=13
0.52 mem=14
0.52 mem=15
0.52 mem=16
0.52 mem=17
0.52 mem=18
0.52 mem=19
0.52 mem=20
0.52 mem=21
0.52 mem=22
0.52 mem=23
0.52 mem=24
0.52 mem=25
0.52 mem=26
0.52 mem=27
0.52 mem=28
0.52 mem=29
0.52 mem=30
0.52 mem=31
0.52 mem=32
0.52 mem=33
0.52 mem=34
0.52 mem=35
0.52 mem=36
0.52 mem=37
0.52 mem=38
0.52 mem=39
0.52 mem=40
0.52 mem=41
0.52 mem=42
0.52 mem=43
0.52 mem=44
0.52 mem=45
0.52 mem=46
0.52 mem=47
0.52 mem=48
0.52 mem=49
0.52 mem=50
0.52 mem=51
0.52 mem=52
0.52 mem=53
0.52 mem=54
0.52 mem=55
0.52 mem=56
0.52 mem=57
0.52 mem=58
0.52 mem=59
0.52 mem=60
0.52 mem=61
0.52 mem=62
0.52 mem=63
0.52 mem=64
0.52 mem=65
0.52 mem=66
0.52 mem=67
0.52 mem=68
0.52 mem=69
0.52 mem=70
0.52 mem=71
0.52 mem=72
0.52 mem=73
0.52 mem=74
0.52 mem=75
0.52 mem=76
0.52 mem=77
0.52 mem=78
0.52 mem=79
0.52 mem=80
0.52 mem=81
0.52 mem=82
0.52 mem=83
0.52 mem=84
0.52 mem=85
0.52 mem=86
0.52 mem=87
0.52 mem=88
0.52 mem=89
0.52 mem=90
0.52 mem=91
0.52 mem=92
0.52 mem=93
0.52 mem=94
0.52 mem=95
0.52 mem=96
0.52 mem=97
0.52 mem=98
0.52 mem=99
0.52 mem=100
0.52 mem=101
0.52 mem=102
0.52 mem=103
0.52 mem=104
0.52 mem=105
0.52 mem=106
0.52 mem=107
0.52 mem=108
0.52 mem=109
0.52 mem=110
0.52 mem=111
0.52 mem=112
0.52 mem=113
0.52 mem=114
0.52 mem=115
0.52 mem=116
0.52 mem=117
0.52 mem=118
0.52 mem=119
0.52 mem=120
0.52 mem=121
0.52 mem=122
0.52 mem=123
0.52 mem=124
0.52 mem=125
0.52 mem=126
0.52 mem=127
0.52 mem=128
0.52 mem=129
0.52 mem=130
0.52 mem=131
0.52 mem=132
0.52 mem=133
0.52 mem=134
0.52 mem=135
0.52 mem=136
0.52 mem=137
0.52 mem=138
0.52 mem=139
0.52 mem=140
0.52 mem=141
0.52 mem=142
0.52 mem=143
0.52 mem=144
0.52 mem=145
0.52 mem=146
0.52 mem=147
0.52 mem=148
0.52 mem=149
0.52 mem=150
0.52 mem=151
0.52 mem=152
0.52 mem=153
0.52 mem=154
0.52 mem=155
0.52 mem=156
0.52 mem=157
0.52 mem=158
0.52 mem=159
0.52 mem=160
0.52 mem=161
0.52 mem=162
0.52 mem=163
0.52 mem=164
0.52 mem=165
0.52 mem=166
0.52 mem=167
0.52 mem=168
0.52 mem=169
0.52 mem=170
0.52 mem=171
0.52 mem=172
0.52 mem=173
0.52 mem=174
0.52 mem=175
0.52 mem=176
0.52 mem=177
0.52 mem=178
0.52 mem=179
0.52 mem=180
0.52 mem=181
0.52 mem=182
0.52 mem=183
0.52 mem=184
0.52 mem=185
0.52 mem=186
0.52 mem=187
0.52 mem=188
0.52 mem=189
0.52 mem=190
0.52 mem=191
0.52 mem=192
0.52 mem=193
0.52 mem=194
0.52 mem=195
0.52 mem=196
0.52 mem=197
0.52 mem=198
0.52 mem=199
0.52 mem=200
0.52 mem=201
0.52 mem=202
0.52 mem=203
0.52 mem=204
0.52 mem=205
0.52 mem=206
0.52 mem=207
0.52 mem=208
0.52 mem=209
0.52 mem=210
0.52 mem=211
0.52 mem=212
0.52 mem=213
0.52 mem=214
0.52 mem=215
0.52 mem=216
0.52 mem=217
0.52 mem=218
0.52 mem=219
0.52 mem=220
0.52 mem=221
0.52 mem=222
0.52 mem=223
0.52 mem=224
0.52 mem=225
0.52 mem=226
0.52 mem=227
0.52 mem=228
0.52 mem=229
0.52 mem=230
0.52 mem=231
0.52 mem=232
0.52 mem=233
0.52 mem=234
0.52 mem=235
0.52 mem=236
0.52 mem=237
0.52 mem=238
0.52 mem=239
0.52 mem=240
0.52 mem=241
0.52 mem=242
0.52 mem=243
0.52 mem=244
0.52 mem=245
0.52 mem=246
0.52 mem=247
0.52 mem=248
0.52 mem=249
0.52 mem=250
0.52 mem=251
0.52 mem=252
0.52 mem=253
0.52 mem=254
0.52 mem=255
0.52 mem=256
0.52 mem=257
0.52 mem=258
0.52 mem=259
0.52 mem=260
0.52 mem=261
0.52 mem=262
0.52 mem=263
0.52 mem=264
0.52 mem=265
0.52 mem=266
0.52 mem=267
0.52 mem=268
0.52 mem=269
0.52 mem=270
0.52 mem=271
0.52 mem=272
0.52 mem=273
0.52 mem=274
0.52 mem=275
0.52 mem=276
0.52 mem=277
0.52 mem=278
0.52 mem=279
0.52 mem=280
0.52 mem=281
0.52 mem=282
0.52 mem=283
0.52 mem=284
0.52 mem=285
0.52 mem=286
0.52 mem=287
0.52 mem=288
0.52 mem=289
0.52 mem=290
0.52 mem=291
0.52 mem=292
0.52 mem=293
0.52 mem=294
0.52 mem=295
0.52 mem=296
0.52 mem=297
0.52 mem=298
0.52 mem=299
0.52 mem=300
0.52 mem=301
0.52 mem=302
0.52 mem=303
0.52 mem=304
0.52 mem=305
0.52 mem=306
0.52 mem=307
0.52 mem=308
0.52 mem=309
0.52 mem=310
0.52 mem=311
0.52 mem=312
0.52 mem=313
0.52 mem=314
0.52 mem=315
0.52 mem=316
0.52 mem=317
0.52 mem=318
0.52 mem=319
0.52 mem=320
0.52 mem=321
0.52 mem=322
0.52 mem=323
0.52 mem=324
0.52 mem=325
0.52 mem=326
0.52 mem=327
0.52 mem=328
0.52 mem=329
0.52 mem=330
0.52 mem=331
0.52 mem=332
0.52 mem=333
0.52 mem=334
0.52 mem=335
0.52 mem=336
0.52 mem=337
0.52 mem=338
0.52 mem=339
0.52 mem=340
0.52 mem=341
0.52 mem=342
0.52 mem=343
0.52 mem=344
0.52 mem=345
0.52 mem=346
0.52 mem=347
0.52 mem=348
0.52 mem=349
0.52 mem=350
0.52
```

29