SAPIENZA Università di Roma
Dipartimento di Informatica

Latch and Flip-flop

Prof. Daniele Gorla

Sequential Nets

SAPIENZA Università di Roma
Dipartimento di Informatica

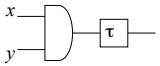
Up to now, we have only considered *acyclic circuits*.

This was due to the implicit assumption that gates are *ideal*, in the sense that they have a zero crossing time.

Under this assumption, the following circuit is meaningless

because the value of y depends by itself (ill-founded definition).

In practice, gates have a non-zero crossing time, tipically modelled thorugh an ideal gate (with zer crossing time) and a delay τ:



This introduces the **time factor** in circuits, che for this reason are called *sequential nets*.

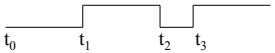
Variables changing over time

SAPIENZA Università di Roma
Dipartimento di Informatica

Hence, boolean variables will not have a simple 0/1 assignment, but such an assignment will change in time.

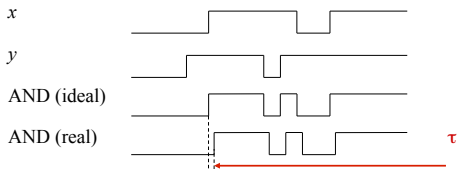
This is represented through a *temporal diagram*, i.e. a graphical representation that shows the variation of values that a variable undergoes over time.

Ex.: x



Similarly, a temporal diagram can represent the evolution in time of a whole sequential net:

Ex.:



Temporal Notation

SAPIENZA Università di Roma
Dipartimento di Informatica

Whereas for combinatorial circuits the expression $y = x \cdot y$ was meaningless, now this makes sense because the leftmost y represents the value of the variable with a delay τ w.r.t. the rightmost y.
→ same line in the temporal diagram, at different moments

For the sake of clarity, we shall write $Y = x \cdot y$, to stress that the first occurrence (Y) represents the value at time $t+\tau$, whereas the second occurrence (y) represents the value at time t.

So, Sequential nets compute boolean functions that vary in time according to:

1. variations of the input values, and
2. Output values of the net in previous time
   → circuits *with memory!!*

Elementary Memory Circuits

The basic sequential circuit is the Flip Flop (FF), a circuit able to store **one bit** (see later on).

Latch SR

This is a preliminary memory circuit with 3 functionalities:

1. Store 1;
2. Store 0;
3. Hold the stored bit.

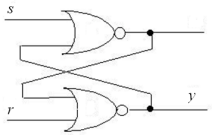
} Codified through 2 bits (s and r)

The three functionalities are called *set*, *reset* and *hold*.

s	r	y	Y
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	-
1	1	1	-

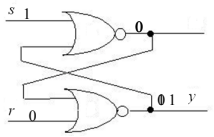
y \ sr	00	01	11	10
0	0	0	-	1
1	1	0	-	1

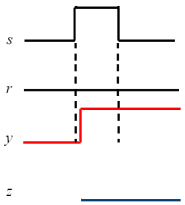
$$Y = \overline{r}(y+s) = \overline{r + (y+s)}$$



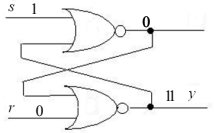
Set status ($s = 1$, $r = 0$)

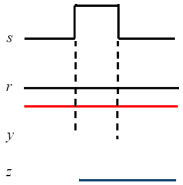
$y = 0$





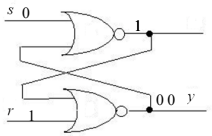
$y = 1$

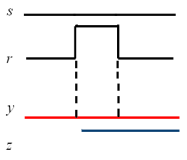




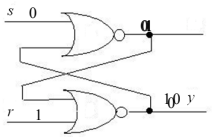
Reset status ($s = 0$, $r = 1$)

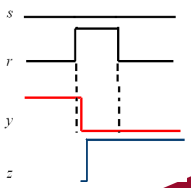
$y = 0$





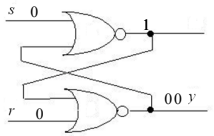
$y = 1$

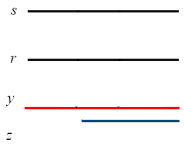




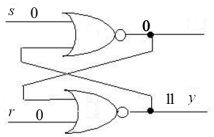
Hold status ($s = 0$, $r = 0$)

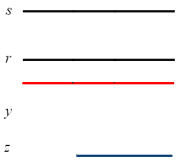
$y = 0$





$y = 1$





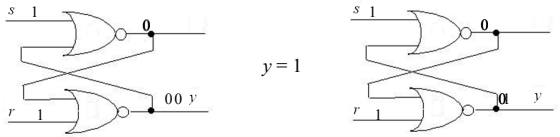
The combination s=r=1 "flattens" the latch

SAPIENZA Università di Roma, Dipartimento di Informatica

In all status seen so far, z is the complement of y (after a proper delay).

This property is lost whenever $s = r = 1$; moreover, in this case both z and y are 0, independently of the initial value of y.

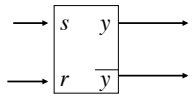
$y = 0$ $\quad$ $y = 1$



$s = r = 1$ is a forbidden input for the latch!!

Latch SR with NOR gates (summing up)

SAPIENZA Università di Roma, Dipartimento di Informatica



Circuit representation

s r	Y
00	y
01	0
10	1
11	-

Characteristic table

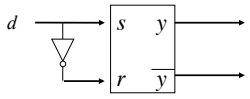
y Y	s r
0 0	0 -
0 1	1 0
1 0	0 1
1 1	- 0

Excitation table

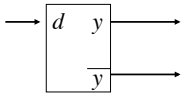
To avoid s = r = 1 (latch D)

SAPIENZA Università di Roma, Dipartimento di Informatica

A way for ensuring that $s = r = 1$ never holds is the following:



This is a new latch, called **latch D** (*delay*), that stores the input d and produces it on the output y after crossing the NOT and NOR gates.



Circuit Representation

d	Y
0	0
1	1

Characteristic table

y Y	d
0 0	0
0 1	1
1 0	0
1 1	1

Excitation fucntion

Use s = r = 1

SAPIENZA Università di Roma, Dipartimento di Informatica

Another way to handle $s = r = 1$ is to add a new functionality to the latch:

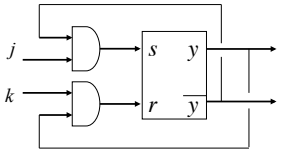
j	k	y	Y
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

like SR

y \ j k	00	01	11	10
0	0	0	1	1
1	1	0	0	1

$$Y = j\bar{y} + \bar{k}y$$

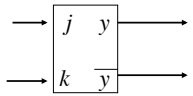
or, by using an SR latch:



Latch JK

SAPIENZA Università di Roma, Dipartimento di Informatica

This is a new latch, called **latch JK**, described by the following:



Circuit representation

$j\,k$	Y
00	y
01	0
10	1
11	$\bar{y}$

Characteristic table

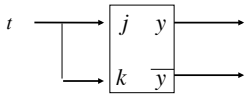
$y\,Y$	$j\,k$
0 0	0 -
0 1	1 -
1 0	- 1
1 1	- 0

Excitation table

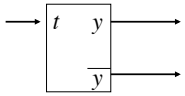
Latch T

SAPIENZA Università di Roma, Dipartimento di Informatica

What happens in a latch JK when $j = k$?



This is a new latch, called **latch T** (*toggle*), that either maintains the stored value ($t = 0$) or complements it ($t = 1$).



Circuit representation

t	Y
0	y
1	$\bar{y}$

Characteristic function

$y\,Y$	t
0 0	0
0 1	1
1 0	1
1 1	0

Excitation function

Synchronous and asynchronous systems

SAPIENZA Università di Roma, Dipartimento di Informatica

Sequential nets can work in a **synchronous** oor in an **asynchronous** way.

- In asynchronous systems, logical circuits change every time that one or more inputs change;
- In synchronous systems, the moment in in which inputs are considered is given by a ”periodic“ signal, called *clock* (a train of square waves):

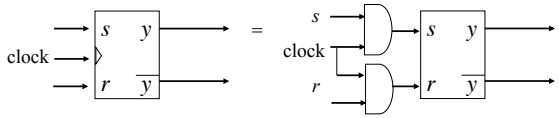
Asynchronous systems have the advantage that istantaneously react to input variations, but this makes their design extremely complex (hence, they are used only when an immediate reaction of the circuit is needed).

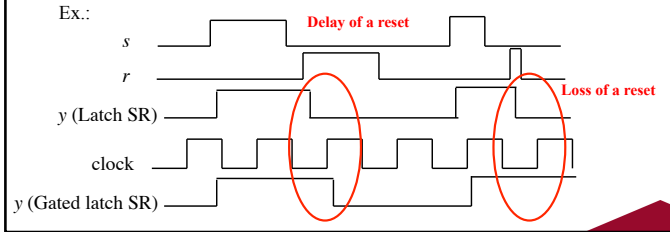
Actual computers almost only exploit **synchronous circuits**!

Gated latch

SAPIENZA Università di Roma, Dipartimento di Informatica

Given a clock signal, we can put it in AND with the latch’s inputs; in this way, inputs are considered only when the clock holds 1.





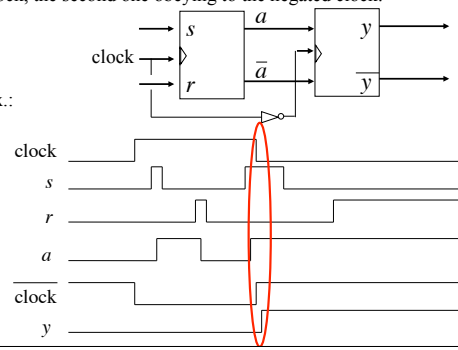
Master-slave Flip-flop

SAPIENZA UNIVERSITÀ DI ROMA
DIPARTIMENTO DI INFORMATICA

Gated latches work well only if the clock holds 1 for a very short time.

A safer solution are the Master-slave Flip-flops (SR, for example), obtained by putting two gated latches in cascade, the first one obeying the clock, the second one obeying to the negated clock:

Ex.:



FF sensible to the descending wave front of the clock