



États Logiques

Exercices Conception numérique



Solution vs. Hints:

Toutes les réponses fournies ici ne sont pas des solutions complètes. Certaines ne sont que des indices pour vous aider à trouver la solution vous-même. Dans d'autres cas, seule une partie de la solution est fournie.

1 | LST - Portes logiques ne fournissant qu'un état

1.1 Circuit à interrupteurs

$$y = \bar{a}$$

lst/one-state-01-01

1.2 Circuit à interrupteurs

$$y = \overline{ab}$$

lst/one-state-01-02

1.3 Circuit à interrupteurs

Multiple possible solution. Minimal with 2 interruptors n-type and 2 interruptors p-type.

lst/one-state-01-03



1.4 Circuit à source ouverte

Missing pull-down resistor on y.

| a | b | c | y |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 |

lst/one-state-02-01

1.5 Alarme

Open-Drain and Open-Source Circuit possible.

lst/one-state-02-02

1.6 Détection de collision

You need either Open-Source or Open-Drain as well as a comparator. The priority depends on the Open-X you've chosen.

lst/one-state-02-03

1.7 Transmission d'information sur un seul fil

Possible with Open-Drain and Open-Source. Only one information can be transmitted at a time.

$$\begin{cases} D_a = '1' \text{ if } a = '0' \text{ and } b = '1' \\ D_b = '1' \text{ if } a = '1' \text{ and } b = '0' \end{cases} \quad (1)$$

lst/one-state-02-04



2 | LST - Portes logiques avec sortie à haute impédance

2.1 Branchement de périphériques en série

Can be done with the help of a DEMUX-1to4.

lst/hiz-01

2.2 Réalisation de fonction avec des circuits haute impédance

Two tri-state buffers are needed.

lst/hiz-02

2.3 Réalisation de fonction avec des circuits à haute impédance

You need one tri-state inverter.

lst/hiz-03

2.4 Détection de collision

You need a comparator and a buffer with tristate functionality. Only one component is enabled at any given time.

lst/hiz-04

2.5 Registre à bus bidirectionnel

The register is written if `wr = '1'` and `cs = '1'` and the register is read if `rd = '1'` and `cs = '1'`. For the chip select get inspired by the exercises *lst_hiz-01*.

lst/hiz-05