



PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE
ESCUELA DE INGENIERÍA
DEPARTAMENTO DE CIENCIA DE LA COMPUTACIÓN

IIC2343 — Arquitectura de Computadores (2020-2)

Ayudantía 2

Lógica Digital (Circuitos)

1 Circuitos Lógicos Combinacionales

Ejercicio 1: I1 2020/1

Un laboratorio está haciendo una investigación en hámsteres a los cuales se les pusieron electrodos para determinar cuándo tienen hambre y/o sueño. Además el laboratorio cuenta con una bodega que puede o no tener comida. Al hámster se le podrá dar comida en su plato, o poner una cama para que duerma, de acuerdo a las siguientes condiciones:

- (I) Si tiene hambre y queda comida en la bodega, entonces se le rellena su plato de comida.
- (II) Si tiene hambre pero no queda comida en la bodega, se le coloca se cama, tenga o no sueño.
- (III) Si tiene hambre y sueño y además queda comida en la bodega, sólo se le rellena su plato.
- (IV) Si sólo tiene sueño, se le coloca su cama.
- (V) Si no tiene hambre ni sueño, el hámster es feliz y no necesita nada.

Debes:

1. Crear la tabla de verdad que siga las condiciones planteadas. Utiliza las letras **H** para hambre, **S** para sueño, **Q** para "queda comida", **P** para "rellenar plato" y **B** para "colocar cama".
2. A partir de la tabla, crea la fórmula lógica para cada salida (**P** y **B**).
3. A partir de la fórmula, crea el circuito lógico combinacional.

Ejercicio 2: I1 2013/1

1. Considere la siguiente fórmula en lógica booleana:

$$Y = (A \wedge B) \vee \neg(A \wedge C) \wedge \neg B$$

(I) Escriba la tabla de verdad correspondiente a la función anterior y diseñe un circuito lógico que la implemente.

(II) A partir de la tabla de verdad, re-escriba la función como conjunto de **ANDs** de las variables (o sus negaciones), combinados mediante **ORs**.

2. Si dos números enteros (con signo), A y B, son sumados (restados) mediante un **adder (subtractor)** de n bits, es posible que el resultado sea incorrecto: la suma de dos números positivos puede ser un número negativo, la suma de dos números negativos puede ser un positivo, etc. Estos errores se conocen como *overflow*.

Para detectar automáticamente estas situaciones, se define una variable binaria **overflow** que toma el valor 1 cuando ocurre el error y 0 cuando no ocurre. Esta variable depende de otras cuatro variables binarias: A_{n-1} (bit más significativo de A), B_{n-1} (bit más significativo de B), S_{n-1} (bit más significativo del resultado de la operación, Op (operación: suma o resta).

Escriba la tabla de verdad para la variable **overflow**, en base a los valores de las otras cuatro variables ya mencionadas.

Ejercicio 3: C1 2019/1

Diseñe un componente de 4 bits que según una señal de control multiplique o divida el valor de 4 bits de entrada por 2. (Sí, solo por 2).

HINT: Los multiplexores existen.