Trabajo Entregable 0 - Introducción al Diseño Lógico

Participantes:
- Federico Goncalves - 03866/5
- Joaquín Guzmán - 03751/4
- Tomás Gamarra - 03852/8

1. Objetivos.

En este informe se busca estudiar la precisión y rango de valores con el que nos permite trabajar la representación en punto fijo de las variables de pendiente (**m**) y ordenada al origen (**b**) de una ecuación de una recta (**y=mx+b).**Así como también poder trabajar con distintas representaciones y poder hacer las operaciones correspondientes. Para ello, se abordarán los siguientes aspectos:

a) Rango de representación y resolución de **m**.
b) Rango de representación y la resolución de **b**.
c) Elección de una representación en punto fijo para **x** e **y** tal que permita representar **m** y **b** sin pérdidas significativas.
d) Rango de representación y resolución de **x** e **y** bajo la representación elegida.
e) Valores a los que debería acotarse **x** para que en casos límites de valores de **m** y **b** no se produzca un overflow de **y** utilizando la representación elegida.

2. A tener en cuenta :

- Asumimos para todas las representaciones de las variables **m**,**b**,**x** e **y** que su interpretación se hace en **CA2**. Para el caso de las variables **m** y **b** que se nos da una representación en punto fijo definida, asumimos que hay un bit más que representa el signo.

a)

Para **m** se adopta una representación **Q(0,15)**.

Rango de representación.

$$R_m = [-(\sum_{i=1}^{15} 2^{-i} + 2^{-15}); \sum_{i=1}^{15} 2^{-i}]$$

$$R_m = [-1; 1 - 2^{-15}] = [-1; 0.9999694824]$$

Resolución.

$$Res_m = 2^{-15}$$

b)

Para **b** se adopta una representación **Q(7,8)**.

Rango de representación.

$$R_b = [-(2^7 - 1 + \sum_{i=1}^{8} 2^{-i} + 2^{-8}); 2^7 - 1 + \sum_{i=1}^{8} 2^{-i}]$$

$$R_b = [-128; 128 - 2^{-8}] = [-128; 127.9960938]$$

Resolución.

$$Res_b = 2^{-8}$$

c)

Para que no haya pérdidas significativas para **m** y **b** siendo representadas en la misma representación que **x** e **y** (**Q(c,d)**) es necesario que el rango de ésta sea por lo menos igual al rango más grande y la resolución sea igual o mayor a la más chica entre las representaciones de m y b.

Rango más grande ------> R_b
Entonces para Q(c,d) ------> c >= 7

Resolución más chica ------> Res_m
Entonces para Q(c,d) ------> d >= 15

Buscamos representar una ecuación de una recta de la forma y=mx+b en donde buscamos que las variables x,y,m y b utilicen una única representación de punto fijo. Llegamos a la conclusión de que para no tener pérdida de cifras significativas en el pase de una representación a otra, necesitaremos de los 32 bits totales un mínimo de **7 bits** asignados a la **parte entera** y un mínimo de **15 bits** asignados a la **parte fraccionaria**.

La asignación de los 10 bits restantes se vuelve dependiente del rango y resolución que se le desee asignar a las variables **x** e **y**. En nuestro caso, elegimos arbitrariamente una división de bits para las partes entera y fraccionaria, quedando finalmente la representación en punto fijo para **x** e **y** : **Q(15,16)** (más el bit de signo).

Ahora hemos de plantear una sistematización para poder pasar cualquier número de representación de **m (Q(0,15))** o **b (Q(7,8))** a la representación elegida para **x** e **y (Q(15,16))**.

A continuación mostramos cómo sería la distribución de los bits para una representación Q(15,16) cualquiera.

Aclaración : Los subíndices de cada bit **b** indican el peso que tiene ese bit en cuanto a potencias de 2 , así como también el subíndice **s** en el bit más significativo (MSB) indica que ese es el **bit de signo**.

$$b_s\ b_{14}\ldots b_0\ ,b_{-1}\ b_{-2}\ b_{-3}\ b_{-4}\ b_{-5}\ b_{-6}\ b_{-7}\ b_{-8}\ b_{-9}\ b_{-10}\ b_{-11}\ b_{-12}\ b_{-13}\ b_{-14}\ b_{-15}\ b_{-16}$$

Caso m: Q(0,15)

Tenemos un valor cualquiera de m tal que sus bits son :

$$x_s\ x_{-1}\ x_{-2}\ x_{-3}\ \ x_{-4}\ x_{-5}\ \ x_{-6}\ x_{-7}\ x_{-8}\ x_{-9}\ \ \ x_{-10}\ x_{-11}\ \ x_{-12}\ x_{-13}\ \ x_{-14}\ x_{-15}$$

x_s=Bit de signo.
x_{-n}=Bit que representa la potencia de 2 negativa n-ésima.

Si ahora solapamos como quedaría la cadena de bits que definimos anteriormente para **m** en una representación Q(15,16) cualquiera , obtendremos lo siguiente :
.

$$b_s\ b_{14}\ldots b_0\ ,b_{-1}\ b_{-2}\ b_{-3}\ b_{-4}\ b_{-5}\ b_{-6}\ b_{-7}\ b_{-8}\ b_{-9}\ b_{-10}\ b_{-11}\ b_{-12}\ b_{-13}\ b_{-14}\ b_{-15}\ b_{-16}$$
$$0_s\ 0_{14}\ldots 0_0\ ,x_s\ \ x_{-1}\ x_{-2}\ x_{-3}\ \ x_{-4}\ x_{-5}\ \ x_{-6}\ x_{-7}\ x_{-8}\ \ x_{-9}\ \ \ x_{-10}\ x_{-11}\ \ x_{-12}\ x_{-13}\ \ x_{-14}\ x_{-15}$$

Entonces la cadena de bits en representación Q(0,15) al querer pasarla a Q(15,16) queda desplazado 1 bit hacia la derecha. Esto puede arreglarse con la operación de desplazamiento a la izquierda por 1 lugar (multiplicación por 2) :

$$b_s\ b_{14}\ldots b_0\ ,b_{-1}\ b_{-2}\ b_{-3}\ b_{-4}\ b_{-5}\ b_{-6}\ b_{-7}\ b_{-8}\ b_{-9}\ b_{-10}\ b_{-11}\ b_{-12}\ b_{-13}\ b_{-14}\ b_{-15}\ b_{-16}$$
$$0_s\ 0_{14}\ldots 0_0\ ,x_s\ \ x_{-1}\ x_{-2}\ x_{-3}\ \ x_{-4}\ x_{-5}\ \ x_{-6}\ x_{-7}\ x_{-8}\ \ x_{-9}\ \ \ x_{-10}\ x_{-11}\ \ x_{-12}\ x_{-13}\ \ x_{-14}\ x_{-15}\ * 2^1$$

$$=$$

$$b_s\ b_{14}\ldots b_0\ ,b_{-1}\ b_{-2}\ b_{-3}\ b_{-4}\ b_{-5}\ b_{-6}\ b_{-7}\ b_{-8}\ b_{-9}\ b_{-10}\ b_{-11}\ b_{-12}\ b_{-13}\ b_{-14}\ b_{-15}\ b_{-16}$$
$$0_s\ \ 0_{14}\ldots x_s\ ,x_{-1}\ x_{-2}\ x_{-3}\ \ x_{-4}\ x_{-5}\ \ x_{-6}\ x_{-7}\ x_{-8}\ x_{-9}\ \ x_{-10}\ x_{-11}\ \ x_{-12}\ x_{-13}\ x_{-14}\ x_{-15}\ 0_{-16}$$

Otro problema que surge ahora es que el bit de signo queda en la posición del bit menos significativo de la parte entera. La solución que planteamos es la siguiente:

$$
\begin{array}{rl}
 & 0_s\ 0_{14}\ 0_{13}\ 0_{12}\ 0_{11}\ 0_{10}\ 0_9\ 0_8\ 0_7\ 0_6\ 0_5\ 0_4\ 0_3\ 0_2\ x_s\ x_0,x_{-1}\ldots x_{-15}\ 0_{-16} \\
+ & 0_s\ 1_{14}\ 1_{13}\ 1_{12}\ 1_{11}\ 1_{10}\ 1_9\ 1_8\ 1_7\ 1_6\ 1_5\ 1_4\ 1_3\ 1_2\ 1_1\ 0_0\ ,0_{-1}\ldots 0_{-15}\ 0_{-16} \\
\hline
 & x_s\ \overline{x}_s\ \overline{x}_s\ \overline{x}_s\ \overline{x}_s\ \overline{x}_s\ \overline{x}_s\ \overline{x}_s\ \overline{x}_s\ \overline{x}_s\ \overline{x}_s\ \overline{x}_s\ \overline{x}_s\ \overline{x}_s\ \overline{x}_s\ ,x_{-1}\ldots x_{-15}\ 0_{-16} \\
\text{AND} & 1_s\ 0_{14}\ 0_{13}\ 0_{12}\ 0_{11}\ 0_{10}\ 0_9\ 0_8\ 0_7\ 0_6\ 0_5\ 0_4\ 0_3\ 0_2\ 0_1\ ,1_{-1}\ldots 1_{-15}\ 1_{-16} \\
\hline
 & x_s\ 0_{14}\ 0_{13}\ 0_{12}\ 0_{11}\ 0_{10}\ 0_9\ 0_8\ 0_7\ 0_6\ 0_5\ 0_4\ 0_3\ 0_2\ 0_s\ ,x_{-1}\ldots x_{-15}\ 0_{-16}
\end{array}
$$

Si ahora volvemos a solapar como quedo la cadena de bits que representaba a **m** en contraste con una representación Q(15,16) cualquiera observamos que quedó expresada correctamente :

$$
\begin{array}{l}
b_s\ b_{14}\ldots b_0\ ,b_{-1}\ b_{-2}\ b_{-3}\ b_{-4}\ b_{-5}\ b_{-6}\ b_{-7}\ b_{-8}\ b_{-9}\ b_{-10}\ b_{-11}\ b_{-12}\ b_{-13}\ b_{-14}\ b_{-15}\ b_{-16} \\
x_s\ 0_{14}\ldots 0_0,x_{-1}\ x_{-2}\ x_{-3}\ x_{-4}\ x_{-5}\ x_{-6}\ x_{-7}\ x_{-8}\ x_{-9}\ x_{-10}\ x_{-11}\ x_{-12}\ x_{-13}\ x_{-14}\ x_{-15}\ 0_{-16}
\end{array}
$$

Como se puede ver sumamos una cadena de bits con los bits que representan la parte entera en 1. Esto genera un acarreo gracias a la suma, con el cual “movemos” el bit de signo al bit más significativo (MSB) y con la operación AND junto con la máscara propuesta convertimos cualquier valor que haya quedado en los bits de parte entera a 0. Es importante aclarar que si el bit de signo es negativo y quisiéramos interpretar la cadena resultante en CA2 , deberíamos convertir los bits que quedaron en 0 de la parte entera a bits en 1 , ya que sino estarían dando un peso a la cadena el cual no corresponde. Sería como hacer una **extensión de signo**.

Caso b: Q(7,8)

Tenemos un valor cualquiera de **b** tal que sus bits son

$$x_s\ x_6\ x_5\ x_4\ x_3\ x_2\ x_1\ x_0,x_{-1}\ x_{-2}\ x_{-3}\ x_{-4}\ x_{-5}\ x_{-6}\ x_{-7}\ x_{-8}$$

Si ahora solapamos como quedaría la cadena de bits que definimos anteriormente para **b** en una representación Q(15,16) cualquiera , obtendremos lo siguiente :

$$
\begin{array}{l}
b_s\ b_{14}\ldots b_0\ ,b_{-1}\ b_{-2}\ b_{-3}\ b_{-4}\ b_{-5}\ b_{-6}\ b_{-7}\ b_{-8}\ b_{-9}\ b_{-10}\ b_{-11}\ b_{-12}\ b_{-13}\ b_{-14}\ b_{-15}\ b_{-16} \\
0_s\ 0_{14}\ldots 0_0\ ,x_s\ x_6\ x_5\ x_4\ x_3\ x_2\ x_1\ x_0\ x_{-1}\ x_{-2}\ x_{-3}\ x_{-4}\ x_{-5}\ x_{-6}\ x_{-7}\ x_{-8}
\end{array}
$$

Notamos que la cadena de bits está desplazada a la derecha 8 lugares. Por lo tanto nuestra primera operación para convertir la cadena de bits en Q(7,8) a Q(15,16) será un desplazamiento a izquierda de 8 lugares (multiplicación por 2^8):

$$b_s\ b_{14}\ldots b_0\ ,b_{-1}\ b_{-2}\ b_{-3}\ b_{-4}\ b_{-5}\ b_{-6}\ b_{-7}\ b_{-8}\ b_{-9}\ b_{-10}\ b_{-11}\ b_{-12}\ b_{-13}\ b_{-14}\ b_{-15}\ b_{-16}$$
$$0_s\ 0_{14}\ldots 0_0\ ,x_s\ \ x_6\ \ x_5\ \ x_4\ \ x_3\ \ x_2\ \ x_1\ x_0\ \ x_{-1}\ \ x_{-2}\ \ x_{-3}\ \ x_{-4}\ \ x_{-5}\ \ x_{-6}\ \ \ x_{-7}\ \ x_{-8} * 2^8$$

$$=$$

$$b_s\ b_{14}\ldots b_8\ \ b_7\ \ b_6\ \ b_5\ \ b_4\ \ b_3\ \ b_2\ \ b_1\ \ b_0\ \ ,b_{-1}\ \ b_{-2}\ \ b_{-3}\ \ \ b_{-4}\ \ b_{-5}\ \ b_{-6}\ \ b_{-7}\ \ b_{-8}\ \ b_{-9}\ldots b_{-16}$$
$$0_s\ 0_{14}\ldots 0_8\ \ x_s\ \ x_6\ \ x_5\ \ x_4\ \ x_3\ \ x_2\ \ x_1\ \ x_0\ \ ,x_{-1}\ \ x_{-2}\ \ x_{-3}\ \ x_{-4}\ \ x_{-5}\ \ x_{-6}\ \ x_{-7}\ \ x_{-8}\ \ 0_{-9}\ldots 0_{-16}$$

Así como en el caso de Q(0,15), nuevamente surge el problema de la posición del bit de signo. En este caso se encuentra en el bit 7 cuando debería encontrarse en el bit más significativo. Las operaciones para desplazar únicamente el bit de signo serán en lógica las mismas que para el caso anterior:

$$\begin{array}{ll} & 0_{15}\ 0_{14}\ 0_{13}\ 0_{12}\ 0_{11}\ 0_{10}\ 0_9\ 0_8\ x_s\ \ x_6\ldots x_{-8}\ 0_{-9}\ldots\ 0_{-16} \\ + & 0_{15}\ 1_{14}\ 1_{13}\ 1_{12}\ 1_{11}\ 1_{10}\ 1_9\ 1_8\ 1_7\ 0_6\ldots 0_{-8}\ 0_{-9}\ldots 0_{-16} \\ \hline & x_s\ \ \overline{x}_s\ \ \overline{x}_s\ \ \overline{x}_s\ \ \overline{x}_s\ \ \overline{x}_s\ \ \overline{x}_s\ \ \overline{x}_s\ \ \overline{x}_s\ x_6\ldots x_{-8}\ 0_{-9}\ldots 0_{-16} \\ \text{AND} & 1_s\ 0_{14}\ 0_{13}\ 0_{12}\ 0_{11}\ 0_{10}\ 0_9\ 0_8\ 0_7\ 0_6\ 0_5\ 0_4\ 0_3\ 0_2\ 0_s\ ,1_{-1}\ldots 1_{-15}\ 1_{-16} \\ \hline & x_s\ 0_{14}\ldots.0_7\ x_6\ x_5\ x_4\ x_3\ x_2\ x_1\ x_0,x_{-1}\ x_{-2}\ x_{-3}\ \ x_{-4}\ x_{-5}\ \ x_{-6}\ x_{-7}\ x_{-8}\ 0_{-9}\ldots.0_{-16} \end{array}$$

De igual forma que **m**, es importante aclarar que si el bit de signo es negativo y quisiéramos interpretar la cadena resultante en CA2 , deberíamos convertir los bits que quedaron en 0 de la parte entera a bits en 1 , ya que sino estarían dando un peso a la cadena el cual no corresponde. Esto es como hacer una **extensión de signo**.
Si ahora volvemos a solapar como quedo la cadena de bits que representaba a **b** en contraste con una representación Q(15,16) cualquiera observamos que quedó expresada correctamente :

$$b_s\ b_{14}\ldots.b_7\ b_6\ b_5\ b_4\ b_3\ b_2\ b_1\ b_0,b_{-1}\ b_{-2}\ b_{-3}\ \ b_{-4}\ b_{-5}\ \ b_{-6}\ \ b_{-7}\ b_{-8}\ b_{-9}\ldots.b_{-16}$$
$$x_s\ 0_{14}\ldots.0_7\ x_6\ x_5\ x_4\ x_3\ x_2\ x_1\ x_0,x_{-1}\ x_{-2}\ x_{-3}\ \ x_{-4}\ x_{-5}\ \ x_{-6}\ x_{-7}\ x_{-8}\ 0_{-9}\ldots.0_{-16}$$

d)

Para **x** e **y** se adopta una representación **Q(15,16)**.

Rango de representación.

$$R_{x,y}=[-\ (2^{15}-1+\sum_{i=1}^{16}2^{-i}+2^{-16})\ ;\ 2^{15}-1+\sum_{i=1}^{16}2^{-i}]$$

$$R_{x,y}=[-\ 32768;\ 32768\ -\ 2^{-16}]=[-\ 32768;\ 32767.99998]$$

Resolución.

$$Res_{x,y}=2^{-16}$$

e)
Determinaremos en base a la representación que elegimos para **x** e **y** (**Q(15,16)**) los **valores máximos** que puede tomar la variable **x** en distintos **casos límites**.
Para hacer esto imponemos valores maximos a las variables **m** y **b** , luego fijamos el valor máximo que podemos representar en la variable **y** en base a los signos de m y de b, finalmente despejamos de la ecuación de la recta el valor máximo de x que puede introducirse para que la variable **y** no se desborde (overflow).

Ecuación de la recta :
$y = mx + b$

Despejando **x** de la ecuación nos quedaría :
$x_{max} = \frac{y-b}{m}$

En base a esta última ecuación calcularemos el valor de **x máximo** que puede tomar la variable para los casos extremos siguientes :

<u>Caso 1:</u>

-El valor más positivo de m → $m = 1 - 2^{-15}$
-El valor más positivo de b → $b = 128 - 2^{-8}$
-El valor más positivo de y → $y = 32768 - 2^{-16}$

Obtenemos como resultado que en estas circunstancias el valor máximo que puede tomar la variable x es → $x_1 = 32641,00002$

<u>Caso 2:</u>

-El valor más negativo de m → $m = -1$

-El valor más positivo de b $\rightarrow b = 128 - 2^{-8}$

-El valor más positivo de y $\rightarrow y = 32768 - 2^{-16}$

Obtenemos como resultado que en estas circunstancias el valor máximo que puede tomar la variable x es $\rightarrow x_2 = -32640,00389$

Caso 3:

-El valor más positivo de m $\rightarrow m = 1 - 2^{-15}$
-El valor más negativo de b $\rightarrow b = -128$
-El valor más negativo de y $\rightarrow y = -32768$

Obtenemos como resultado que en estas circunstancias el valor máximo que puede tomar la variable x es $\rightarrow x_3 = -32640,00389$

Caso 4:

-El valor más negativo de m $\rightarrow m = -1$
-El valor más negativo de b $\rightarrow b = -128$
-El valor más negativo de y $\rightarrow y = -32768$

Obtenemos como resultado que en estas circunstancias el valor máximo que puede tomar la variable x es $\rightarrow x_4 = 32640$

En cuanto al control que llevaremos de los valores máximos que puede tomar la variable **x** en el programa trabajaremos con un intervalo definido en base a estos valores límites calculados anteriormente. Este intervalo irá desde el número menos negativo que obtuvimos ($x_2 = -32640,00389$) hasta el valor menos positivo que obtuvimos ($x_4 = 32640$). Utilizando esta cota podremos asegurarnos que el programa será preciso en el cálculo del valor de **y** evitando desbordamiento en el resultado final, lo que generaría errores.

Aclaración en cuanto al desarrollo del código:
A la hora de resolver los problemas de los enunciados *f* y *g*,habíamos implementado las funciones para esos casos específicos, lo cual funcionaba, pero al querer reutilizar las funciones para el inciso h esto era poco eficiente. Por lo que decidimos hacer una librería de la cual podamos usar las funciones para todos los incisos. A su vez decidimos generalizar los métodos para que funcionen para cualquier representación **Q(c,d)** en 32 bits , enviando como parámetros a **c** y **d** a las distintas funciones. Sin embargo esto complicó un poco el desarrollo del código, lo cual podríamos haber evitado si hubiésemos hecho las funciones para los casos específicos de representaciones que se nos solicitaba (Q(0,15) y Q(7,8)). De igual forma, vimos una oportunidad de intentar desafiarnos en los algoritmos y desarrollarlo de esta manera la cual sería más que eficiente y nos dará la oportunidad de en un futuro trabajar con distintas representaciones.