DEPARTAMENTO
DE COMPUTACION
Facultad de Ciencias Exactas y Naturales - UBA

Práctica 1: Apunte sobre Overflow

Números enteros

Segundo Cuatrimestre 2020

Organización del Computador I
DC - UBA

Operaciones representables

Overflow es la interpretación de la representabilidad del resultado de una operación aritmética sobre elementos de tamaño fijo.

- Por ejemplo, tenemos ocho bits para el dato con un bit de signo
s	a_6	a_5	a_4	a_3	a_2	a_1	a_0
- Son $2^7(128)$ elementos para cada lado de la región representable de -128 a -1 para los números negativos y de 0 a 127 para los positivos

Operaciones sobre enteros de tamaño fijo

Vamos a trabajar con **sumas** y **restas**, con 8 bits tenemos un rango de enteros representables entre -128 y 127.

Observación
Para sumas de distinto signo vale:

$$a \in [-128 \ldots -1] \land b \in [0 \ldots 127]$$

$$\text{entonces } a + b \in [-128 \ldots 126]$$

Lo que significa que la suma de distinto signo es cerrada para enteros de 8 bits. El caso donde los signos están al revés es igual. Lo mismo pasa con restas de igual signo, porque restar dos números de igual signo es como sumar números de distinto signo $a - b = a + (-b)$.

Operaciones sobre enteros de tamaño fijo

Vamos a ver qué es lo que esperamos para cada operación. Observemos que para un dominio de suma (la resta es igual) podemos tener un rango de -256 a 254 que resultan de sumar los dos números más positivos o más negativos representables en 8 bits complemento a dos. Son valores representables en 9 bits, con lo cual si tenemos un problema de representabilidad va a estar en el bit más alto, ya que la operación desborda y utiliza el bit de signo para representar la magnitud.

Descomponiendo en casos simbólicos

Vamos a analizar dos casos, el de la suma y el de la resta, observando solamente el bit más significativo y declarando qué condiciones consideramos de overflow. Vimos que sumar dos números de signos opuestos siempre es representable, lo mismo para la resta de números de igual signo.

Descomponiendo en casos simbólicos

Vamos a suponer que tenemos n bits para representar a nuestros números en complemento a dos y notamos a la operación como $a \cdot b = s$.

op	a_{n-1}	b_{n-1}	s_{n-1}
$+$	0	0	0
$+$	1	1	1
$+$	$1 - a_{n-1}$	$1 - b_{n-1}$	x
$-$	0	1	0
$-$	1	0	1
$-$	b_{n-1}	a_{n-1}	x

De simbólico a extensivo

Vamos a declarar que hay overflow cuando no se cumplen éstas condiciones. Primero armemos tablas de verdad por extensión.

op	a_{n-1}	b_{n-1}	s_{n-1}	**V**	op	a_{n-1}	b_{n-1}	s_{n-1}	**V**
$+$	0	0	0	0	$-$	0	0	0	0
$+$	0	0	1	1	$-$	0	0	1	0
$+$	0	1	0	0	$-$	0	1	0	1
$+$	0	1	1	0	$-$	0	1	1	1
$+$	1	0	0	0	$-$	1	0	0	0
$+$	1	0	1	0	$-$	1	0	1	1
$+$	1	1	0	1	$-$	1	1	0	0
$+$	1	1	1	0	$-$	1	1	1	0

De simbólico a extensivo

De aquí ya pueden armar un circuito por forma normal conjunta o disjunta.

op	a_{n-1}	b_{n-1}	s_{n-1}	**V**	op	a_{n-1}	b_{n-1}	s_{n-1}	**V**
$+$	0	0	0	0	$-$	0	0	0	0
$+$	0	0	1	1	$-$	0	0	1	0
$+$	0	1	0	0	$-$	0	1	0	1
$+$	0	1	1	0	$-$	0	1	1	1
$+$	1	0	0	0	$-$	1	0	0	0
$+$	1	0	1	0	$-$	1	0	1	1
$+$	1	1	0	1	$-$	1	1	0	0
$+$	1	1	1	0	$-$	1	1	1	0

Otro enfoque

De aquí podríamos haber notado que los valores legales vinculan s_{n-1} con a_{n-1} en relación a comparar a_{n-1} con b_{n-1}.

op	a_{n-1}	b_{n-1}	s_{n-1}
$+$	0	0	0
$+$	1	1	1
$+$	$1 - a_{n-1}$	$1 - b_{n-1}$	x
$-$	0	1	0
$-$	1	0	1
$-$	b_{n-1}	a_{n-1}	x

Otro enfoque

Y con la observación anterior podríamos haber escrito esta tabla por extensión sobre parámetros simbólicos.

op	$a_{n-1} = b_{n-1}$	$a_{n-1} = s_{n-1}$	**V**	op	$a_{n-1} = b_{n-1}$	$a_{n-1} = s_{n-1}$	**V**
$+$	0	0	0	$-$	0	0	0
$+$	0	1	0	$-$	0	1	1
$+$	1	0	1	$-$	1	0	0
$+$	1	1	1	$-$	1	1	0

Otro enfoque

Otra vez, pueden armar un circuito de aquí, pero las entradas a su vez son salidas de una comparación, que se puede impelemntar con compuertas xor negadas.

op	$a_{n-1} = b_{n-1}$	$a_{n-1} = s_{n-1}$	**V**	op	$a_{n-1} = b_{n-1}$	$a_{n-1} = s_{n-1}$	**V**
$+$	0	0	0	$-$	0	0	0
$+$	0	1	0	$-$	0	1	1
$+$	1	0	1	$-$	1	0	0
$+$	1	1	1	$-$	1	1	0

Conclusión

Lo importante, al final del día, es tener herramientas para poder construir los mecanismos que nos hacen falta. Las herramientas pueden ser conceptuales (abstracción y definición de parámetros de entrada y salida) o mecánicas (construcción de tablas de verdad), pero a veces tomarse un poco más de tiempo en el planteo conceptual hace que la parte mecánica tenga sentido.