

Software & Hardware Interface (Data Section)

Jiaming Xu

2023.9.20

Why binary digits

- the first commercial computer did offer decimal arithmetic
- decimal signal is hard for hardware
- electrical signal is recognized as on and off much more efficiently

Why main memory in binary sizes

- binary size: main memory (DRAM, SRAM)
 - information is preserved by binary numbers
 - addresses in loads and stores are binary numbers
- decimal size: web package, disk

Why Tmin -2147483647-1

- $-X = -(X)$

1. find data type to represent 2147483648
2. negates it

ISO C90		ISO C99	
Decimal	Hexadecimal	Decimal	Hexadecimal
int long unsigned unsigned long	int unsigned long unsigned long	int long long long	int unsigned long unsigned long long long unsigned long long

Figure 1: **Data types for representing integer constants.** According to the language version and format (decimal or hexadecimal), the data type for a constant is given by the first type in the appropriate list that can represent the value.

Word Size	ISO C90		ISO C99	
Expression	-2147483648	0x80000000	-2147483648	0x80000000
32	unsigned	unsigned	long long	unsigned
64	long	unsigned	long	unsigned

Figure 2: **Data types resulting from constant expressions for $TMin_{32}$.** According to the language version and format (decimal or hexadecimal), we can get three different data types for the two expressions, including cases where the value is positive.

Integer types (signed)	signed char	Same size as char. At least 8 bits.
	<i>signed short int</i>	Not smaller than char. At least 16 bits.
	<i>signed int</i>	Not smaller than short. At least 16 bits.
	<i>signed long int</i>	Not smaller than int. At least 32 bits.
	<i>signed long long int</i>	Not smaller than long. At least 64 bits.

Why unsigned

- memory addresses naturally start at 0 and continue to the largest address
- negative addresses make no sense

ALU

- learn it in *Digital Logic Design*

Handle integer overflow

- C/C++/Java is not everything
- Ada/Fortran notification
- even an exception to stop the program in MIPS
 - add/subtract ✓
 - multiply x

Separate floating-point registers

- twice register w/o using up extra bits of instruction
- twice the register bandwidth
- create a separate set of data transfer instructions

Reference

- CS:APP Web Aside DATA:TMIN: Writing TMin in C
- Computer Organization And Design: The Hardware/Software Interface (David a. Patterson, John L. Hennessy) Chapter 1,2,3