SEMINAR 3

ARITHMETICS AND INTRO TO ARM

UNSIGNED INTEGERS ARE OBVIOUS, AREN'T THEY? :)

INTRODUCING SIGN

- SIGN BIT + ABSOLUTE VALUE
- INVERSE CODE
- COMPLEMENT CODE

INTRODUCING SIGN

- SIGN BIT + ABSOLUTE VALUE
- INVERSE CODE
- COMPLEMENT CODE

SIZE SPECIFIERS

TYPE	SIZE ACCORDING TO STANDARD (BITS)
CHAR	AT LEAST 8
SHORT	AT LEAST 16
INT	AT LEAST 16
LONG	AT LEAST 32
LONG LONG	AT LEAST 64

WHEN PRECISE SIZE IS REQUIRED, USE STDINT.H

REPRESENTATION OF SIGNED INTS IN C

NOW: REPRESENTATION IS NOT DEFINED, BUT CAST TO UNSIGNED INT WORKS AS IF SIGNED INTEGER IS REPRESENTED USING COMPLEMENT CODE

SINCE C23: REPRESENTED USING COMPLEMENT CODE

UNSIGNED OVERFLOW IS MODULAR ARITHMETICS
SIGNED OVERFLOW IS UNDEFINED BEHAVIOUR

DEMO

COMPILER OPTIMIZES COMPARISON

IEEE 754

FLOAT - 32 BITS, 1 BIT SIGN, 8 BITS EXPONENT, 23 BITS FRACTION
DOUBLE - 64 BITS, 1 BIT SIGN, 11 BITS EXPONENT, 52 BITS FRACTION

$$x = (-1)^S \cdot 2^{E-B} \cdot \left(1 + \sum_i m_i \cdot 2^{-i}\right)$$

WHERE B = 127 FOR FLOAT AND B = 1024 FOR DOUBLE

SPECIAL VALUES

- E = 0XFF...FF, M = 0 IS INFINITY
- S = 1, E = 0, M = 0 IS NEGATIVE ZERO
- E = 0 IS DENORMALIZED NUMBER
- E = 0XFF..FF, M =/= 0 IS NAN

ARMV8

COMPLETE INSTRUCTION SET COMPUTER

VS

RESTRICTED INSTRUCTION SET COMPUTER

REGISTERS

IN ARMV8 PROCESSOR WE HAVE

31 GENERAL-PURPOSE 64 BIT REGISTERS X0-X30, LOWER HALF CAN BE ADDRESSED USING W0-W30

XZR/WZR ARE ALWAYS ZERO

SP IS STACK POINTER

REGISTERS (MORE)

PC IS PROGRAM COUNTER (INACCESSIBLE DIRECTLY)

SOME REGISTERS FOR FLOATING-POINT NUMBERS AND VECTOR REGISTERS

REGISTERS (WRAP-UP)

SPSR - SAVED PROCESS STATUS REGISTER FOR FLAGS

- CARRY
- OVERFLOW
- NEGATIVE
- ZERO

CALLING CONVENTION

X0-X7 FOR ARGUMENTS (RETURN VIA X0)
X8-X18 TEMPORARY CALLER-SAVED REGISTERS
X19-X28 ARE CALLEE-SAVED REGISTERS
X29 IS FRAME POINTER
X30 IS RETURN ADDRESS

DATA SEGMENTS

.TEXT FOR EXECUTABLE CODE
.DATA FOR INITIALIZED STATIC VARIABLES
.BSS FOR UNINITIALIZED STATIC VARIABLES
.RODATA FOR CONSTANTS
.DEBUG FOR DEBUG INFO

DEMO