

# User-Visible Architectural State

- Program counter (`%rip`).
  - Special register. Not directly accessible from user-mode code.
- Integer register file (`R[ ]`).
  - 16 named general-purpose registers storing 64-bit values.
  - Distinct name space from memory. Accessed by name.
- Condition flags.
  - Multiple single-bit registers containing status information about the most recently executed arithmetic/logical instructions.
  - Used to support branching.
- Memory (`M[ ]`).

# C and Intel Terminologies

C declaration	Intel term	ASM suffix	Size (B)
<code>char</code>	Byte	<code>b</code>	1
<code>short</code>	Word	<code>w</code>	2
<code>int</code>	Double word	<code>l</code>	4
<code>long</code>	Quad word	<code>q</code>	8
<code>char *</code>	Quad word	<code>q</code>	8
<code>float</code>	Single precision	<code>s</code>	4
<code>double</code>	Double precision	<code>l</code>	8

# x86 Assembler Formats

- Two incompatible formats of writing assembly-language instructions.
  - ATT format: Source operands first, destination last. Register names have the `%` prefix.
  - Intel format: Destination first, followed by source operands. Register names don't have the `%` prefix.
- We will use SASM (a cross-platform IDE) to abstract away from these inessential differences. SASM supports multiple assemblers.
  - NASM (Netwide Assembler) <https://nasm.us>
  - MASM (Microsoft Macro Assembler) <https://docs.microsoft.com/en-us/cpp/assembler/masm/microsoft-macro-assembler-reference?view=msvc-160>
  - GAS (GNU Assembler) <https://www.gnu.org/software/binutils/>
  - FASM (Flat Assembler) <http://flatassembler.net/download.php>

# General-Purpose Registers in x86

63:32	31:16	15:8	7:0
%rax	%eax	%ax	%al
%rbx	%ebx	%bx	%bl
%rcx	%ecx	%cx	%cl
%rdx	%edx	%dx	%dl
%rsi	%esi	%si	%sil
%rdi	%edi	%di	%dil
%rbp	%ebp	%bp	%bpl
%rsp	%esp	%sp	%spl
%r8	%r8d	%r8w	%r8b
%r9	%r9d	%r9w	%r9b
%r10	%r10d	%r10w	%r10b
%r11	%r11d	%r11w	%r11b
%r12	%r12d	%r12w	%r12b
%r13	%r13d	%r13w	%r13b
%r14	%r14d	%r14w	%r14b
%r15	%r15d	%r15w	%r15b