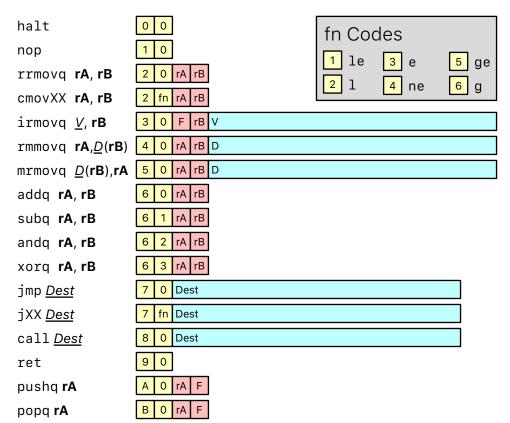
CSCI 370: Computer Architecture

Y86-64 Reference

Instruction Format



Registers

ID	Enc	Usage	
%rdi	7	arg1	
%rsi	6	arg2	
%rdx	2	arg3	eq
%rcx	1	arg4	saved
%r8	8	arg5	er-
%r9	9	arg6	са11е
%rax	0	return	Са
%r10	Α	general	
%r11	В	general	
%rbx	3	general	1
%r12	С	general	saved
%r13	D	general	-sa
%r14	Е	general	lee
%rsp	4	stack ptr	callee-
%rbp	5	base ptr	J
	F	no reg	

Status Conditions

AOK	1	Normal	
HLT	2	Halt Encountered	
ADR	3	Bad Address	
INS	4	Invalid Instruction	

HCL Y86-64 Hardware Registers

stage	register(s)	description
Fetch	icode,ifun	Read instruction byte
	rA,rB	Read register byte
	valC	Read constant word
	valP	Compute next PC
Decode	valA,srcA	Read operand A
	valB,srcB	Read operand B
Execute	valE	Perform ALU operation
	cnd	Set/Use Condition Code
Memory	valM	Memory Read/Write
Writeback	dstE	Write back ALU result
	dstM	Write back Mem result
PC Update	PC	Update PC

Y86-64 Data Example

Assembly Translation Example

```
/* find number of elements in null-terminated list */
long len(long* a) {
  long len;
  for (len = 0; a[len]; ++len)
  return len;
len:
    irmovq $1, %r8
                            # Constant 1
    irmovq $8, %r9
                            # Constant 8
    irmovq $0, %rax
                            \# len = 0
    mrmovq (%rdi), %rdx
                            \# val = *a
    and %rdx, %rdx
                            # Test val
    je Done
                            # If zero, goto Done
Loop:
    addq %r8, %rax
                            # len++
    addq %r9, %rdi
                            # a++
    mrmovq (%rdi), %rdx
                            \# val = *a
    andq %rdx, %rdx
                            # Test val
    jne Loop
                            # If !0, goto Loop
Done:
    ret
```