RTN (Register Transfer Notation)

- Provides a formal means of describing machine structure and function
- Is at the "just right" level for machine descriptions
- Does not replace hardware description languages
- Can be used to describe *what* a machine does (an abstract RTN) without describing *how* the machine does it
- Can also be used to describe a particular hardware implementation (a concrete RTN)

RTN (cont'd.)

- At first you may find this "meta description" confusing, because it is a language that is used to describe a language
- You will find that developing a familiarity with RTN will aid greatly in your understanding of new machine design concepts
- We will describe RTN by using it to describe SRC

Some RTN Features— Using RTN to Describe a Machine's Static Properties

Static Properties

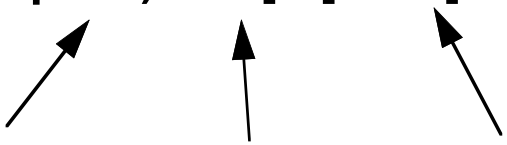
- Specifying registers
  - IR⟨31..0⟩ specifies a register named "IR" having 32 bits numbered 31 to 0
- "Naming" using the := naming operator:
  - op⟨4..0⟩ := IR⟨31..27⟩ specifies that the 5 msbs of IR be called op, with bits 4..0
  - Notice that this does not create a new register, it just generates another name, or "alias," for an already existing register or part of a register

Using RTN to Describe Dynamic Properties

Dynamic Properties

- **Conditional expressions:**

(op=12) → R[ra] ← R[rb] + R[rc]: ; defines the add instruction



"if" condition "then" RTN Assignment Operator

This fragment of RTN describes the SRC add instruction. It says, "when the op field of IR = 12, then store in the register specified by the ra field, the result of adding the register specified by the rb field to the register specified by the rc field."

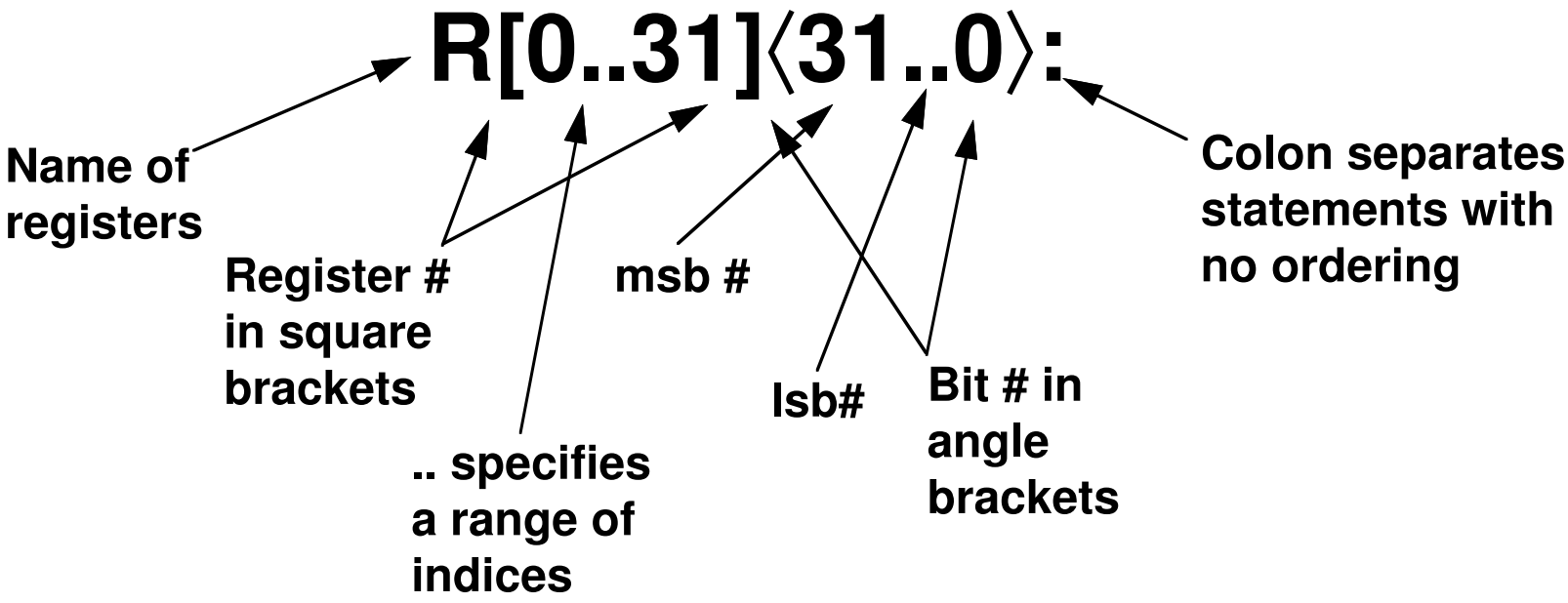
Using RTN to Describe the SRC (Static) Processor State

Processor state

PC⟨31..0⟩:	program counter (memory addr. of next inst.)
IR⟨31..0⟩:	instruction register
Run:	one bit run/halt indicator
Strt:	start signal
R[0..31]⟨31..0⟩:	general purpose registers

RTN Register Declarations

- General register specifications shows some features of the notation
- Describes a set of 32 32-bit registers with names R[0] to R[31]



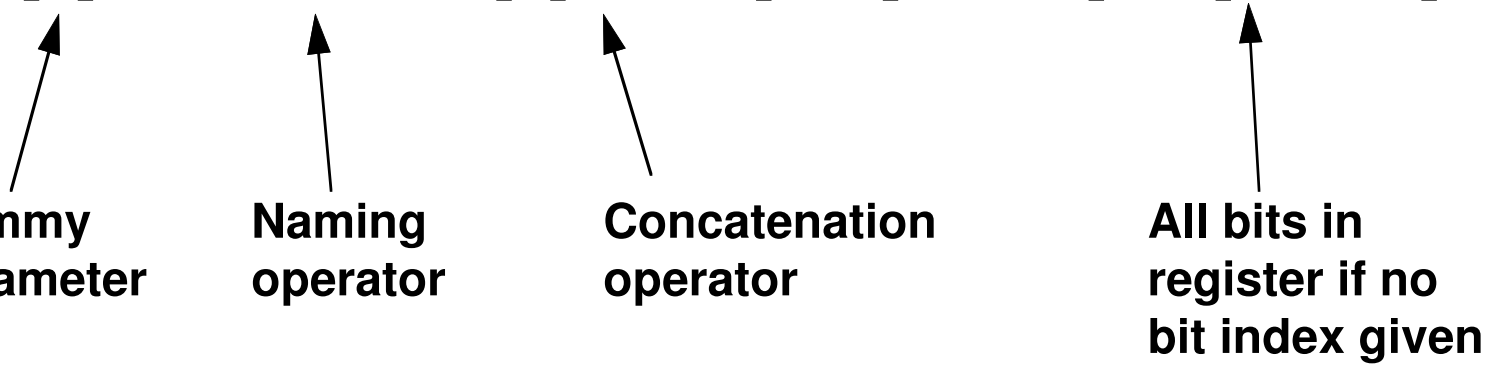
Memory Declaration: RTN Naming Operator

- Defining names with formal parameters is a powerful formatting tool
- Used here to define word memory (big-endian)

Main memory state

Mem[0..2^{32} - 1]⟨7..0⟩: 2^{32} addressable bytes of memory

M[x]⟨31..0⟩:= Mem[x]#Mem[x+1]#Mem[x+2]#Mem[x+3]:



RTN Instruction Formatting Uses Renaming of IR Bits

Instruction formats

op⟨4..0⟩ := IR⟨31..27⟩:	operation code field
ra⟨4..0⟩ := IR⟨26..22⟩:	target register field
rb⟨4..0⟩ := IR⟨21..17⟩:	operand, address index, or branch target register
rc⟨4..0⟩ := IR⟨16..12⟩:	second operand, conditional test, or shift count register
c1⟨21..0⟩ := IR⟨21..0⟩:	long displacement field
c2⟨16..0⟩ := IR⟨16..0⟩:	short displacement or immediate field
c3⟨11..0⟩ := IR⟨11..0⟩:	count or modifier field

Specifying Dynamic Properties of SRC: RTN Gives Specifics of Address Calculation

Effective address calculations (occur at runtime):

$disp\langle 31..0\rangle := ((rb=0) \rightarrow c2\langle 16..0\rangle$ {sign extend}: displacement
$(rb\neq 0) \rightarrow R[rb] + c2\langle 16..0\rangle$ {sign extend, 2's comp.}): address
$rel\langle 31..0\rangle := PC\langle 31..0\rangle + c1\langle 21..0\rangle$ {sign extend, 2's comp.}: relative address

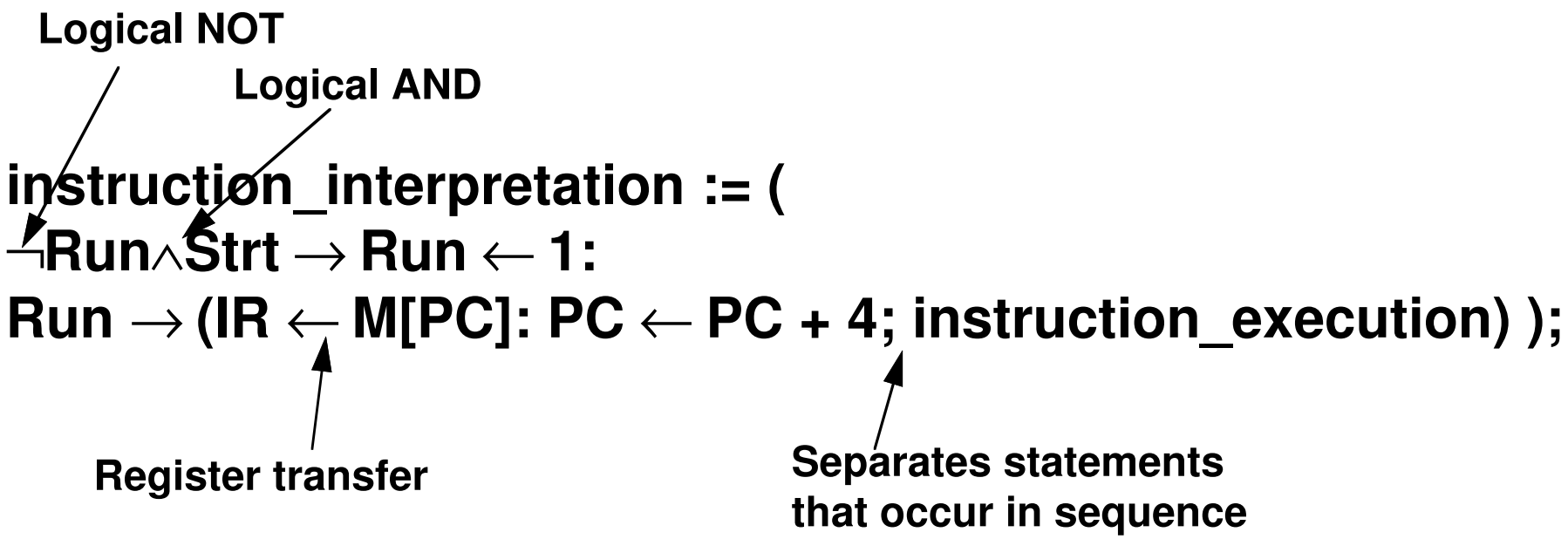
- Renaming defines displacement and relative addresses
- New RTN notation is used
  - condition $\rightarrow$ expression means <u>if</u> condition <u>then</u> expression
  - modifiers in { } describe type of arithmetic or how short numbers are extended to longer ones
  - arithmetic operators (+ - * / etc.) can be used in expressions
- Register R[0] cannot be added to a displacement

Detailed Questions Answered by the RTN for Addresses

- What set of memory cells can be addressed by direct addressing (displacement with rb=0)
  - If c2⟨16⟩=0 (positive displacement) absolute addresses range from 00000000H to 0000FFFFH
  - If c2⟨16⟩=1 (negative displacement) absolute addresses range from FFFF0000H to FFFFFFFFH
- What range of memory addresses can be specified by a relative address
  - The largest positive value of C1⟨21..0⟩ is 2^{21}-1 and its most negative value is -2^{21}, so addresses up to 2^{21}-1 forward and 2^{21} backward from the current PC value can be specified
- Note the difference between rb and R[rb]

Instruction Interpretation: RTN Description of Fetch-Execute

- Need to describe actions (not just declarations)
- Some new notation



RTN Sequence and Clocking

- In general, RTN statements separated by : take place during the same clock pulse
- Statements separated by ; take place on successive clock pulses
- This is not entirely accurate since some things written with one RTN statement can take several clocks to perform
- More precise difference between : and ;
  - The order of execution of statements separated by : does not matter
  - If statements are separated by ; the one on the left must be complete before the one on the right starts