## CSE 378, 06wi – Lecture 2 Main Points Introduction to the MIPS ISA

January 6, 2006

Hardware (as exposed to software)

- 1+31 = 32 registers (register 0 always contains 0)
- $2^{30}$  words of main memory (1 word = 4 bytes = 32 bits)  $\circ 2^{32}$  bytes of main memory
- a program counter (PC)
  - "points to" the next instruction to execute

**Instruction Set** (expressed in assembler)

•	add	rd, rs, rt	Example:	add	\$8, \$4, \$5
•	sub	rd, rs, rt		sub	\$8, \$5, \$6
•	addi	rt, rs, immed		addi	\$8, \$4, 100
•	lw	rd, offset(base)		lw	\$8, 20(\$4)
•	SW	rt, offset(base)		SW	\$8, 24(\$5)

**Instruction Encoding** (binary – what's stored in memory)

•	add (R-format)							
	31 2	6 25 21	20 16	15 11	10 6	5 0		
	SPECIAL			-	0	ADD		
	000000	15	ĸ	14	00000	100000		
	6	5	5	5	5	6		

• addi (I-format)

31	26	25 21	20 16	15 0
ADDI		1.4.4	10000	1. M 10 10 10 10 10 10 10 10 10 10 10 10 10
001000	ē	rs	п	immediate
6		5	5	16

• lw (I-format)

31	26 Dilliat	25	21 20 1	6 15	0
	LW 100011	base	rt	offset	
	6	5	5	16	

## Compiling

• Going from a high-level language representation of a program to an equivalent sequence of assembler instructions