Support of MIPS Assembly Instruction Set

Legend: Supported instructions - blue, Unsupported instructions (to be implemented as part of LAB5) - black

Arithmetic Instructions			
Instruction	***Example***	***Meaning***	***Comments***
add	add $1,$2,$3	$1=$2+$3	
subtract	sub $1,$2,$3	$1=$2-$3	
add immediate	addi $1,$2,100	$1=$2+100	
Multiply (without overflow)	mul $1,$2,$3	$1=$2*$3	The destination register is only 32 bits, where for the two source registers, only the lower half word (16-bit) content matters

Logical and shift Instructions			
Instruction	***Example***	***Meaning***	***Comments***
and	and $1,$2,$3	$1=$2&$3	Bitwise AND
or	or $1,$2,$3	$1=$2|$3	Bitwise OR
xor	xor $1,$2,$3	$1=$2^$3	Bitwise XOR
and immediate	andi $1,$2,100	$1=$2&100	Bitwise AND with immediate value
or immediate	ori $1,$2,100	$1=$2|100	Bitwise OR with immediate value
xor immediate	xori $1,$2,100	$1=$2^100	Bitwise XOR with immediate value
shift left logical	sll $1,$2,10	$1=$2<<10	Shift left by a constant number of bits
shift right logical	srl $1,$2,10	$1=$2>>10	Shift right by a constant number of bits

Data Transfer Instructions			
Instruction	***Example***	***Meaning***	***Comments***
move	move $1,$2	$1=$2	Pseudo-instruction (provided by MARS Assembler, not processor!) Copy from register to register.
load address	la $1, label	$1= label address	Pseudo-instruction (provided by MARS Assembler, not processor!) Loads the computed address of a label (not its data content) into a register
load immediate	li $1,100	$1=100	*Pseudo-instruction* (provided by MARS Assembler, not processor!) Loads an immediate value into a register
load word	lw $1,100($2)	$1=Memory[$2+100]	
store word	sw $1,100($2)	Memory[$2+100]=$1	
load upper immediate	lui $1,100	$1=100x$2^{16}$	Load constant into upper 16 bits. Lower 16 bits are set to zero.

Conditional Branch Instructions			
Instruction	***Example***	***Meaning***	***Comments***
branch on equal	beq $1,$2,100	if($1==$2) go to PC+4+100	Test if registers are equal
branch on not equal	bne $1,$2,100	if($1!=$2) go to PC+4+100	Test if registers are not equal
branch on greater or equal than	bge $1,$2,100	if($1>=$2) go to PC+4+100	*Pseduo-instruction*
branch on less than	blt $1,$2,100	if($1<$2) go to PC+4+100	*Pseduo-instruction*

Comparison Instructions			
Instruction	***Example***	***Meaning***	***Comments***
set on less than	slt $1,$2,$3	if($2<$3)$1=1; else $1=0	Test if less than. If true, set $1 to 1. Otherwise, set $1 to 0.
set on less than immediate	Slti $1,$2,100	if($2<100)$1=1; else $1=0	Test if less than. If true, set $1 to 1. Otherwise, set $1 to 0.

Unconditional Jump Instructions			
Instruction	***Example***	***Meaning***	***Comments***
jump	j 1000	go to address 1000	Jump to target address
jump register	jr $ra	go to return address stored in $ra	For switch, procedure return
jump and link	jal 1000	1. Save the procedure return address $ra=PC+4 2. Go to a procedure call address, which starts at address 1000	Use when making procedure call. This saves the return address in $ra