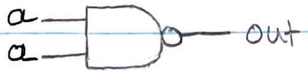
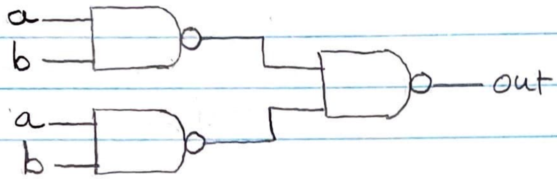
David Giacobbi
CPSC 260: Assignment #3

1.) NAND Diagram: !a



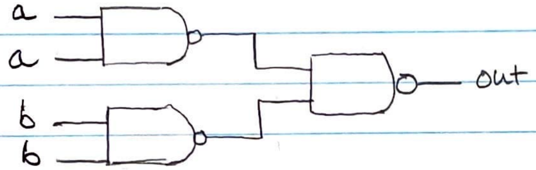
a	a	out	
0	0	1	✓
1	1	0	✓

2.) NAND Diagram: a && b



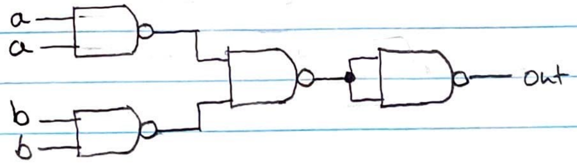
a	b	out	
0	0	0	✓
0	1	0	✓
1	0	0	✓
1	1	1	✓

3.) NAND Diagram: a || b



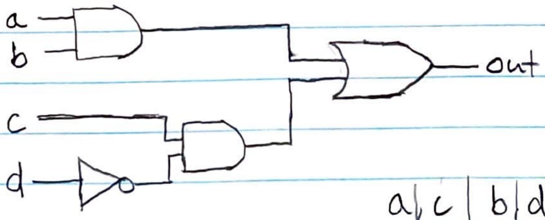
a	b	out	
0	0	0	✓
0	1	1	✓
1	0	1	✓
1	1	1	✓

4.) NAND Diagram: a NOR b



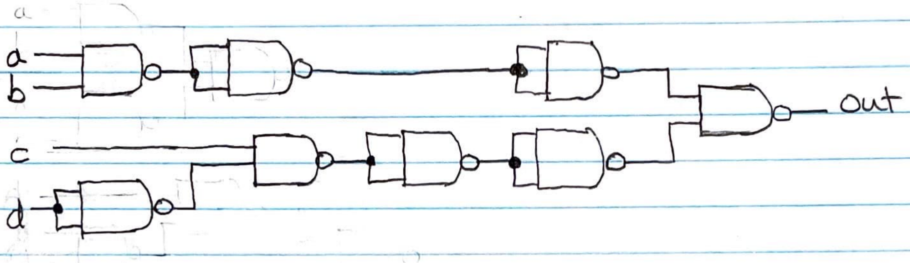
a	b	out	
0	0	1	✓
0	1	0	✓
1	0	0	✓
1	1	0	✓

5.) AND, OR, NOT Diagram: (a && b) || (c&& !d)



a	c	b	d	(a && b)	(c&& !d)	out
0	0	0	0	0	0	0
0	1	0	1	0	0	0
1	0	1	0	0	1	1
1	1	1	1	1	0	1

6.) NAND Diagram: (a && b) || (c && !d)



	Part 2:		C-code translation
hw_test1: (-Og)	mov	$0x0, %eax	// move zero to return
	ret		return 0;
hw_test1 (-O0)	push	%rbp	// start main
	mov	%rsp, %rbp	// initialize x and y
	movl	$0x7, -0xc(%rbp)	int x = 7;
	movl	$0x2, -0x8(%rbp)	int y = 2;
	mov	-0xc(%rbp)	// initialize z & calculate
	imul	-0x8(%rbp)	int z = x * y;
	mov	%eax, -0x4(%rbp)	// update z with calculation
	mov	$0x0, %eax	// move zero to return
	pop	%rbp	// end main
	ret		return 0;

	Assembly		C-code translation
hw-test2	mov	$0x0, %eax	// move zero to return
(-Og)	ret		return 0;
hw-test2	push	%rbp	// start main
(-O0)	mov	%rsp, %rbp	// initialize x and y
	movl	$0x7, -0x8(%rbp)	int x = 7;
	movl	$0x0, -0x4(%rbp)	int y = 0;
	mov	-0x8(%rbp), %eax	// move x val into if-statemen
	and	$0x1, %eax	// move 2 into if-statement
	test	%eax, %eax	if (x % 2 == 0)
	jne	40112b <main+0x25>	// Check if true / perform
	movl	$0x2, -0x4(%rbp)	y = 2;
	jmp	401132 <main+0x2c>	else
	movl	$0x1, -0x4(%rbp)	y = 1;
	mov	$0x0, %eax	// Move zero to return
	pop	%rbp	// end main
	ret		return 0;