

Implementación

1 se define los registros de cada variable

2 con la variable sintética MOV que representar la variable OR se inicializan los registros

3 se inicia con la función comparar CMP que compara los registros esa función es sintética y representa a la función SUPCC.

4 utilizamos la función branch que permite ejecutar los saltos a las etiquetas.

5 se realizar la operaciones que corresponde al condicional.

6 las funciones de las etiquetas es que ayuda a realizar los saltos con el fin de que se realice la operación correspondiente.

Direccion	op	RD	op3	Rs1	i	unused(zero)	rs2
0x0000	10	10001	000010	00000	1	0000000010100	
0x0004	10	10010	000010	00000	1	0000000001011	
0x0008	10	10010	010010	00000	0	00000000 10001	
Direccion	op	a	cound	op2	disp22		
0x000C	0	1	0001	0010	0000000000000000000011		
Direccion	op	RD	op3	Rs1	i	unused(zero)	rs2
0x0010	10	10001	000000	10001	1	000000000000001	
Direccion	op	a	cound	op2	disp22		
0x0014	0	1	1000	0010	0000000000000000000010		
Direccion	op	RD	op3	Rs1	i	unused(zero)	rs2
0x0018	10	10001	000100	10001	1	000000000000001	
0x001C	10	1000	000000	10001	0	00000000 10010	
Direccion	op	RD	op2	IM22			
0x0020	0	00000	100	0000000000000000000000			

Conclusión

Se continúa la práctica, de instrucciones que se ha explicado y aplicado en clase.

Se conoce las nuevas instrucciones como el branch y CMP para las comparaciones y saltos de líneas ayudando a la simplificación.

Para las instrucciones branch se utiliza el formato op2, para las demás líneas siguen utilizando el formato op3.

```
int main(){
    int i=20;
    int j=11;
    if(i!=j){
        i++;
    }
    else{
        i--;
    }
    %L1=i
    %L2=j
    MOV 20, %L1
    MOV 11, %L2
    cmp %L1,%L2
    BE i, FALSE
    ADD %L1,1,%L1
    BA i EXIT
    ADD %L1,%L2,%L3
    FALSE
    SUB %L1,1,%L1
    BA i EXIT
    ADD %L1,%L2,%O0
    EXIT NOT
```

Reemplazar
LOGOTIPO

DANIELA DELGADO
GALEANO
Cc:1088327948