Estructura del Computador - 66.70
MICROCÓDIGO

```
0: R[IR] <- AND(R[PC], R[PC]); READ;
1: DECODE;

/SETHI
X:    R[RD]<- LSHIFT10(IR);
      GOTO 2047;

/CALL
X: R[15] <- AND(R[PC], R[PC]);
X+1:  R[TEMP0] <- ADD(R[IR], R[IR]);
X+2:  R[TEMP0] <- ADD(R[TEMP0], R[TEMP0]);
X+3:  R[PC] <- ADD(R[PC], R[TEMP0]);
      GOTO 0;

/ADDCC ANDCC ORCC ORNCC SRL JMPL
X:    IF R[IR[13]] THEN GOTO X+2;
X+1:  R[RD] <- OPERACION(R[RS1], R[RS2]);
      GOTO 2047;
X+2:  R[TEMP0] <- SIMM13[R[IR]];
X+3:  R[RD] <- OPERACION(R[RS1], R[TEMP0]);
      GOTO 2047;

/JMPL y ADDCC usan de operación ADD y en vez de SIMM13 usan SEXT13
/JMPL GOTO 0;

/LD
X:    R[TEMP0]<- ADD(R[RS1], R[RS2]);
      IF R[IR[13]] THEN GOTO X+2;
X+1:  R[RD]<- ADD(R[TEMP0], R[TEMP0]);
      READ; GOTO 2047;
X+2:  R[TEMP0]<- SEXT13(R[IR]);
X+3:  R[TEMP0]<- ADD(R[RS1], R[TEMP0]);
      GOTO X+1;

/ST
X:    R[TEMP0]<- ADD(R[RS1], R[RS2]);
      IF R[IR[13]] THEN GOTO X+2;
X+1:  R[IR]<- RSHIFT5(R[IR]);
      GOTO 40;
```

```
40:   R[IR]<- RSHIFT5(R[IR]);
41:   R[IR]<- RSHIFT5(R[IR]);
42:   R[IR]<- RSHIFT5(R[IR]);
43:   R[IR]<- RSHIFT5(R[IR]);
44:   R[0]<- AND(R[TEMP0], R[TEMP0]);
      WRITE; GOTO 2047;
X+2:  R[TEMP0]<- SEXT13(R[IR]);
X+3:  R[TEMP0]<- ADD(R[RS1], R[TEMP0]);
      GOTO X+1;

2047: R[PC]<- INCPC(R[PC], R[PC]);
      GOTO 0;
```

Las líneas 0, 1 y 2047 son las que representan en su mayoría el ciclo de Fetch para ejecutar una instrucción. Igualmente se debe recordar que no todas las instrucciones necesitan GOTO 2047.

La dirección de memoria X se consigue resolviendo el DECODE que se escribe de la siguiente manera para cada tipo de instrucción:

1 - OP (2 bits) - OP3/OP2 (6 bits) - 0 - 0

En estos 11 bits se encuentra la dirección de la primera microinstrucción de una instrucción.

Observar que las instrucciones ADDCC, ANDCC, ORCC, ORNCC, JMPL y SRL tienen la misma base para su microcódigo, cambia según la operación y lo que se especificó más arriba. Lo mismo con LD y ST, marcado en negrita la diferencia que tienen.