Descripcion de bloques Procesador:

Registros:

IR: Instruction register, de longitud 16 bits, recibe las instrucciones de DataRom, y las almacena al recibir una señal de control, su salida son todos los bits de su instrucción, estos de dirigen a IR Logic.

IRL (Codename OSO): Instruction Register Low, de longitud 16 bits, recibe la instrucción del IR al recibir una senal de control. Su salida son todos los bits de su instrucción y se dirigen a IRLogic.

PC: Program counter, contador de programa, contador de longitud 11 bits, que se auto incrementa al recibir una señal de control, puede cambiar su valor por el valor recibido de IR Logic, Accumulator 2, MAR, o MDR de acuerdo a otra señal de control. Envía su valor a Accumulator 1, MAR y MDR.

IRL Logic: Bloque que recibe los dos Instruction Register, tanto el actual como el IR low, de acuerdo a la información que se necesite en ese estado permite el paso de la información de IR o IRL de acuerdo un bit de control. Envía datos a PC, MAR, MDR, los Registros, el Counter Register y los Acumuladores.

CONTROL: Bloque central del procesador, contiene la máquina de estados del sistema, envía señales de activación y cambia de estados de acuerdo a AHPL, se comunica con todos los bloques del sistema, con buses de acuerdo al diagrama de bloques.

ACCUMULATOR1: Registro acumulador 1, registro sobre el que se realizan la mayoría de operaciones, recibe datos de acumulador, IR, IRL, el banco de registros y MDR y sobre escribe su valor con los datos recibidos, de acuerdo a una señal de control, envía su contenido como salida a los registros, MDR y la ALU.

ACCUMULATOR2: Registro acumulador 2, registro sobre el que se realizan la mayoría de operaciones, recibe datos de acumulador, IR, IRL, el banco de registros, ALU y MDR y sobre escribe su valor con los datos recibidos, de acuerdo a una señal de control, envía su contenido como salida a los registros, MDR y la ALU

ALU: Bloque de operaciones aritmético lógicas, opera los datos que recibe en su entrada A y B o solamente la de su entrada A, el bloque ALU realiza Suma, Resta, Multiplicación, AND OR y XOR, el valor de A lo recibe de Acumulador 1 y el de B lo recibe de acumulador 2, el resultado de la operación lo almacena en acumulador 2 y la operación la selecciona de acuerdo a una señal recibida de control

SR: Status Register, registro donde se almacenan las banderas de: Zero, Negative, Negative, Carry, Overfflow, end multiply , Interruption enable, Periferico 1, Periferico 2 y Periferico 3, de la siguiente manera:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| SR Bit | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|  | Negative | Zero | Carry | Overflow | EndMultiply | InterruptionEnable | Prioridad1 | Prioridad2 | Prioridad3 |

Estos bits pueden ser modificados por el control, la ALU y los perifericos

CR: Count register, registro de conteo, donde se modifica el valor de conteo máximo, cuenta con un habilitador proveniente de control, al alcanzar el valor de conteo envía una bandera en 1 a control.

MAR: Memory Adress Register, Registro de direcciones de memoria que recibe una dirección de control, o IR y la envía a Output Selector o MDR

SP: Stack Pointer, contador que almacena la dirección actual del stack, se auto incrementa o decrementa con un enable y una señal del control y envía su valor al output select.

OUTPUT SELECTOR: Bloque de multiplexado y almacenamiento, que almacena una dirección de salida, que se envía fuera del procesador al sistema externo de manejo de memorias, recibe las direcciones del SP, el PC, y el MAR y selecciona cual enviar de acuerdo a una señal de control.

MDR: Bloque de multiplexado y almacenamiento, recibe los datos de los acumuladores 1 y2, el SR, el Bidir y el PC. Reescribiendo su valor con una señal de control. Adicionalmente envía los datos almacenados al Bidir y los acumuladores.

BIDIR: Registro bidireccional de entrada y salida de datos, de acuerdo a una señal de control recibe o envía datos desde o hacia el bidir o la memoria.

REGISTERS: Bloque de registros, contiene 31 registros distribuidos de la siguiente forma:

|  |  |
| --- | --- |
| Reg0 | MAR |
| Reg1 | SP |
| Reg2 | SR |
| Reg3 | PcMod |
| Reg4 | Zeros |
| Reg5 | -1 cte |
| Reg6 | 1 cte |
| Reg7 | Id Periferico 1 |
| Reg8 | Id Periferico 2 |
| Reg9 | Id Periferico 3 |
| Reg10 | Id Periferico 4 |
| Reg11 | RPG [0] |
| Reg12 | RPG [1] |
| Reg13 | RPG [2] |
| Reg14 | RPG [3] |
| Reg15 | RPG [4] |
| Reg16 | RPG [5] |
| Reg17 | RPG [6] |
| Reg18 | RPG [7] |
| Reg19 | RPG [8] |
| Reg20 | RPG [9] |
| Reg21 | RPG [10] |
| Reg22 | RPG [11] |
| Reg23 | RPG [12] |
| Reg24 | RPG [13] |
| Reg25 | RPG [14] |
| Reg26 | RPG [15] |
| Reg27 | RPG [16] |
| Reg28 | RPG [17] |
| Reg29 | RPG [18] |
| Reg30 | RPG [19] |
| Reg31 | RPG [20] |