Cruz Leandro Daniel 311113316
Mata Mota Luis Valentín 311239816

Jupiter Ace

En cada fascículo se habla de la descripción de una computadora, en el fascículo que nos toco encontramos el modelo de la Jupiter Ace. Algo en lo que se hace mucho énfasis es en que el lenguaje de programación que maneja no es Basic. Muchos desarrolladores de la época y hasta la fecha coinciden en el martirio que es Basic, aunque otros nos estén tanto de acuerdo. Nosotros no lo conocemos, sin embargo nos dimos a la tarea de leer un poco de él, y aunque parece ser un lenguaje potente, es un poco tedioso y con una estructura un tanto compleja.

Jupiter Ace viene con lenguaje Forth, incluso el mismo artículo muestra una comparación de un programa en ambos lenguajes. A simple viste se puede ver la simplicidad de este lenguaje y por qué puede ser un éxito que una computadora maneje este lenguaje más simplificado y con programas mucho más ligeros.

Algo en lo que también se hace énfasis es que el lenguaje Forth permite crear funciones y agregarlas al diccionario lo que le da al desarrollador grandes ventajas para el diseño de sus programas.

La computadora Jupiter Ace muestra muchas similitudes con la Spectrum ya que de hecho fue el mismo equipo el que desarrollo ambas computadoras, con diferencias al parecer mínimas pero que en general hacen que la computadora sea otra y al parecer mejor, aun cuando la mejora con más atención y por la que parece ser que tuvo gran aceptación fue simplemente cambiar el lenguaje.

Algo que puede llamar la atención es como estas computadoras, Jupiter Ace y las de Sinclair promocionan mucho la portabilidad ya que deecho la salida de datos es través de un televisor.

Este equipo cuenta con una salida en blanco y negro por un modulador RF.
Al ser un diseño heredado de los modelos ZX-80 y ZX-81 de Sinclair, pose una gran compatibilidad con accesorios como el incremento de memoria. No es indispensable ya que como ya mencionamos el lenguaje Forth tiene la particularidad de al ser mas sencillo, crear programas mucho más ligeros, pero si algo ha demostrado la historia de la computación, los desarrolladores siempre estamos buscando más y más memoria.

Localizaciones de Memoria

Este tema lo escogimos porque consideramos que es muy importante aparte que tiene que ver con temas que se han visto en clase como por ejemplo manejo de memoria, bloque de control, registros y bus.

Bien, en este tema se empieza por decir que toda instrucción y dato que procesara la computadora esta situada en alguna dirección de memoria del ordenador. Parece no sonar muy difícil ya que como se dijo toda instrucción tiene una dirección y solo basta saber qué dirección para empezar a procesar la información, pero aquí la duda que surgió es ¿Cómo la computadora iba a interpretar la información? ¿Cómo sabría que es una instrucción o un dato que debe guardar o almacenar?

Reconociendo los códigos

Aquí se nos menciona todo de una manera muy técnica, nos explica la definición de instrucción en modo binario. Nos da un tipo ejemplo de que va a pasar cuando la computadora interprete como instrucción y no como un dato a una instrucción ya determinada en código binario.

Nos explica todo el proceso o etapas que van pasando desde el principio. Que pasa al haber leído la instrucción, de como es decodificada por el bloque de control y como es que se alinean las patillas de dirección (algo de lo cual no teníamos conocimiento) para empezar a leer la instrucción.

Después se nos dice que esta instrucción es un método para direccionar una localización de memoria para recuperar un dato y que es tan solo uno de los métodos de manejo de memoria del cual el programador puede disponer al momento de programar. Menciona que este procedimiento es uno de los ocupados por el microprocesador Z80 (que también se vio en clase) y que en llega a variar en los diferentes chips de microprocesadores, pero siempre siguiendo el mismo principio para direccionar.

Transfiriendo números

Aquí se nos menciona un proceso llamado activación que se puede decir que para este ejemplo de invertir el contenido en el registro de direcciones, la CPU asigna a un dispositivo que se está activo cuando las demás localizaciones de memoria están inactivas. Menciona que cuando se desea decodificar unas pequeñas líneas de dirección se pueden usar chips de compuertas lógicas simples y que este proceso se ocupa cuando se quiere la decodificación de selección para dispositivos de I/O. Pero cuando se quiere la decodificación de muchas localizaciones de dirección es preferible que se lleve a cabo en el chip de memoria ya que la complejidad del circuito aumenta mucho.