

Controladores de lógica programable (PLC)

Estructuras de software

Los PLC cada vez se parecen cada vez más a un computador, no obstante en sus inicios cumplía objetivos de lógica combinacional pura.

El programa es una secuencia de líneas de contactos similar a las de WLC. donde las funciones se refieren a entradas / salidas físicas del PLC .

De lo expuesto se deduce que el PLC es un sistema con lenguajes orientados a la automatización, como ejemplo se describe un set de instrucciones genérico del mismo, con las siguientes prestaciones :

Instrucciones lógicas : AND, OR, NOT, XOR, SET, RESET

Instrucciones de módulos de programación : Temporizadores, contadores, registros de desplazamiento, comparadores.

Instrucciones de control: Control de marcha, condiciones de restart, forzado de variables, inhibición de salidas, saltos condicionales.

Instrucciones matemáticas: Suma, resta, multiplicación, división.

Instrucciones de comparación : Comparación de bits, bytes. igualdad, mayor que, menor que.

Instrucciones de translación : Translación de datos en memoria

Instrucciones de conversión : decimal, binario, BCD

Ciclo de programa(Tiempo de ciclo)

Una de las características fundamentales del PLC es el funcionamiento cíclico de la CPU, y un parámetro de las prestaciones es su **tiempo de scan (tiempo de ciclo)**, considerado como el necesario para ejecutar 1K instrucciones, pero como no todas las instrucciones son iguales en cuanto a tiempo de ejecución se refiere, se adopto como parámetro el tiempo de ejecución de 1 K de memoria vacía.

Básicamente todos los PLC's tienen dos modos de funcionamiento. El primero denominado PROGRAM MODE, permite programar y transferir el programa de la aplicación desde la consola de programación al PLC.

El segundo permite al PLC controlar el proceso, ejecutando el programa y se lo denomina RUN MODE.

Además algunos disponen de un tercer modo denominado MONITOR, con el cual se puede monitorear el PLC corriendo en modo RUN,

Un autómatas típico puede leer mil instrucciones en 6[mseg] y emplea de 5 a 10 [mseg] para actualizar las variables, por lo tanto puede emplearse para controlar

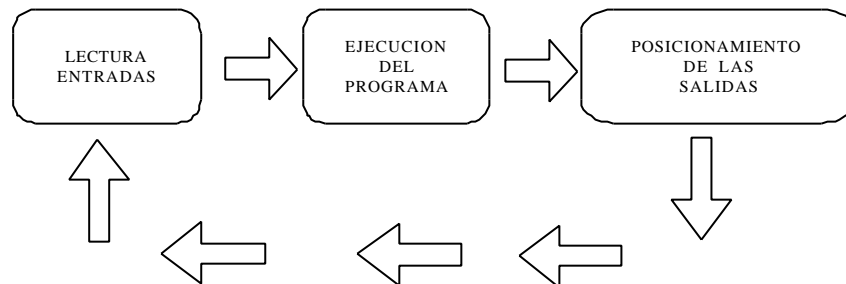
procesos relativamente rápidos. La suma del tiempo de programa y el tiempo de actualización de las variables conforman el tiempo de ciclo, y obviamente es necesario que

tiempo de evolución del proceso >> tiempo de ciclo

Se pueden señalar cuatro modalidades de funcionamiento de acuerdo al equipo y la tecnología con que está construido:

1

- Lectura de las entradas
- Ejecución del programa
- Posicionamiento de las salidas



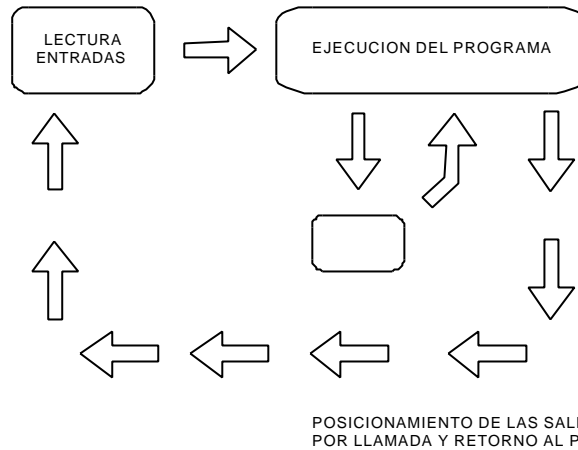
Este suele ser el caso más frecuente, la primera operación llevada a cabo es la lectura de los dispositivos conectados en las entradas del PLC, que será acompañada por la construcción de una imagen de memoria de los estados de los mismos (Registro Imagen de las Entradas), éste contenido permanece inalterable durante todo el ciclo. Una vez actualizadas todas las entradas, la CPU comienza a ejecutar el programa. En la medida que el programa se va ejecutando, la CPU colocará los resultados en un registro denominado (Registro Imagen de las salidas).

2

- Lectura de las entradas
- Ejecución del programa y posicionamiento contemporáneo de las salidas

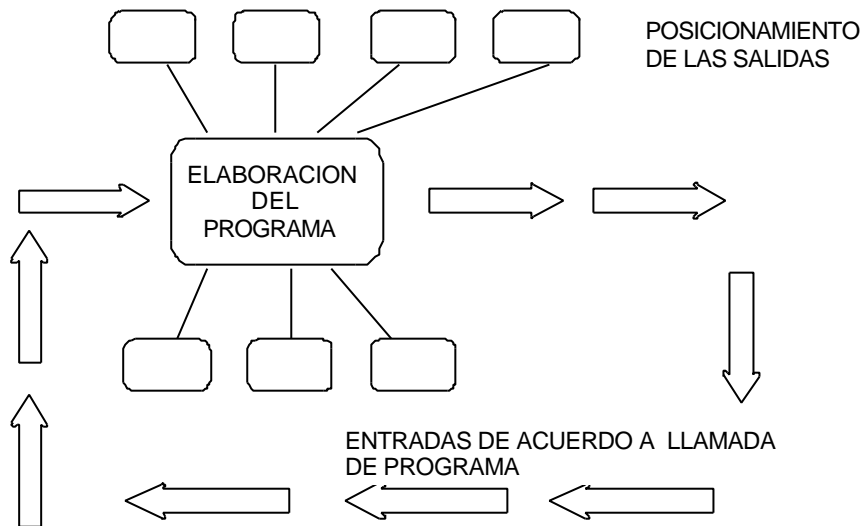
El ciclo comienza con la lectura de las entradas, y el correspondiente registro de datos, pasando posteriormente a la ejecución del programa.

La actualización de las entradas no es consecutiva a la ejecución de todas las instrucciones que componen al programa, sino de acuerdo como se va resolviendo el mismo.



3

- *Ejecución del programa conjuntamente con la lectura de las entradas, siguiendo las llamadas del mismo programa y posicionamiento inmediato de las salidas*

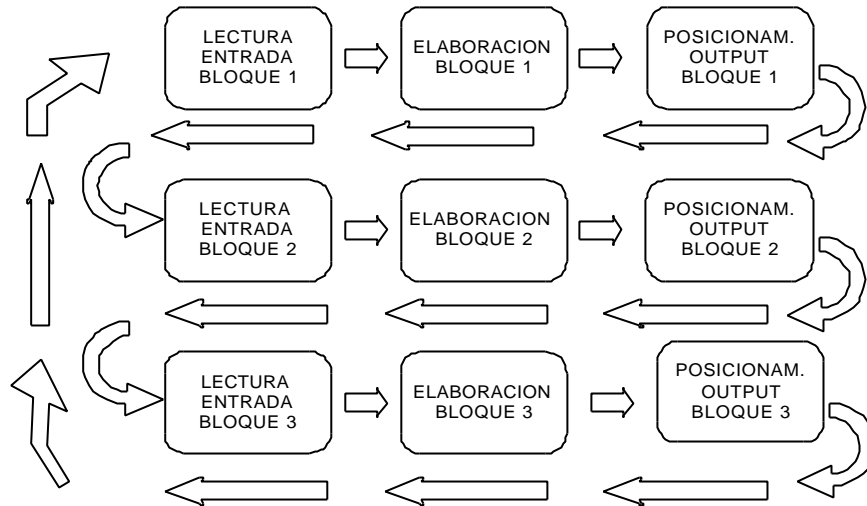


El programa se ejecuta desde su inicio , la interrogación de los dispositivos de entrada se realiza a partir del módulo correspondiente, las salidas se actualizan una en continuación de la otra, en la medida que se ejecutan las instrucciones, esta modalidad no contempla el registro imagen en memoria.

4

- *Subdivisión del programa de bloques y dentro de cada bloque, antes de pasar al siguiente, actualización de las entradas involucradas, ejecución del programa almacenado en el bloque y posicionamiento de las salidas correspondientes.*

El programa se descompone en bloques, los ciclos de ejecución se suceden uno a continuación del otro.



En general para todos los casos, el programa es barrido secuencialmente desde el principio hasta el fin, recomenzando nuevamente en forma de loop cerrado, al día de hoy existen gran variedad de funciones lógicas, aritméticas, especiales que permiten al PLC cubrir amplios espectros de la automatización industrial.

Toda vez que la CPU requiere información inherente a una entrada/salida, se remitirá al registro imagen para verificar el estado de la variable correspondiente.

El tiempo que emplea la CPU para actualizar el registro imagen varía en función del número de dispositivos conectados al sistema. Puede oscilar entre 3 o 4 [µS] hasta algunas decenas de [µS].

Un programa escrito en lenguaje de contactos (ladder) está formado por bloques lógicos conteniendo cada uno de los contactos y de las salidas .

Un bloque lógico está constituido de elementos conectados entre si por conexiones serie y paralelo dispuestas entre dos líneas verticales que representan la polaridad usada en los esquemas eléctrico

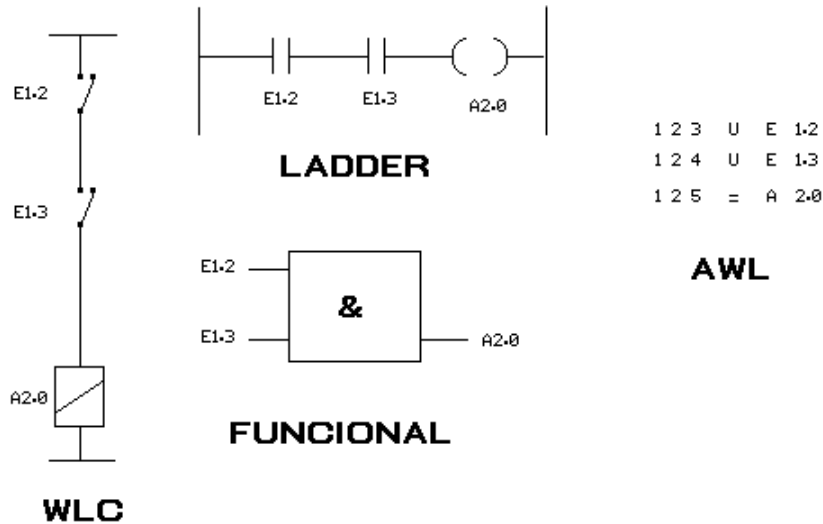


Estos símbolos son los más usados en lenguaje ladder y de ellos derivan muchos más que fueron incorporándose a medida que los PLC's fueron ampliando su gama de prestaciones.

Existen también funciones especiales, temporizadores, contadores, interrupciones, subrutinas, etc

Estos símbolos son los más usados en lenguaje ladder y de ellos derivan muchos más que fueron incorporándose a medida que los PLC's fueron ampliando su gama de prestaciones.

Existen también funciones especiales, temporizadores, contadores, interrupciones, subrutinas, etc



Además del ladder existen otros lenguajes de programación que representan las funciones simbólicamente con mnemónicos, o con lista de instrucciones. Aquí un fragmento de programa en ladder clásico (KOP):

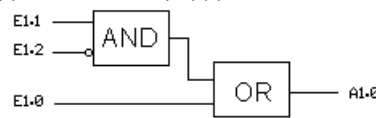


Puede ser expresado en lista de instrucciones (AWL) como :

```

U   E1.1
UN  E1.2
O   E1.0
=   A1.0
    
```

y con diagramas funcionales (FUP):



Tomando como ejemplo la aplicación inicial del montacargas de tres posiciones que fuera resuelto por medio de lógica cableada; el primer paso es definir la entrada/salida de datos, las entradas serán todos los pulsadores, finales de carrera y switchs que informen al sistema las decisiones que el mismo debe adoptar, y las salidas serán los actuadores por medio de los cuales el sistema puede ejercer el control sobre el proceso.

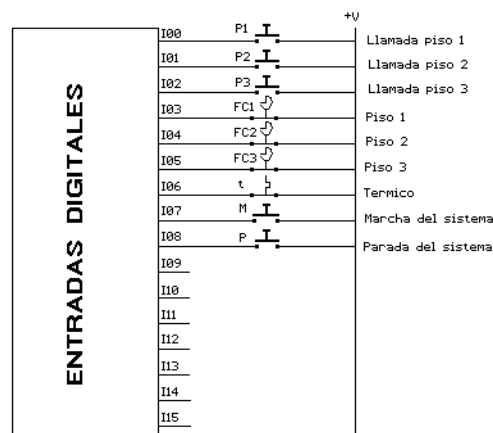
Las variables deben ser ingresadas al PLC por medio de algún módulo de la gama del controlador seleccionado, y las salidas del mismo modo, por medio de actuadores que dispongan de bobinas, serán controlados como un simple contactor.

Suponer una máquina con los siguientes I/O

- FC1, FC2, FC3** finales de carrera
- P1, P2, P3** Pulsadores
- M, P** marcha/parada del sistema
- t** relé térmico del motor

y las salidas a controlar :

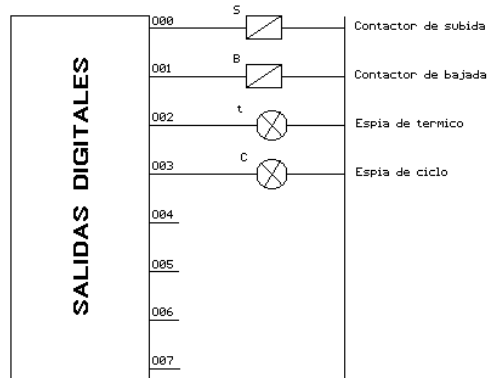
- S** contactor de subida de la máquina
- B** contactor de bajada de la máquina



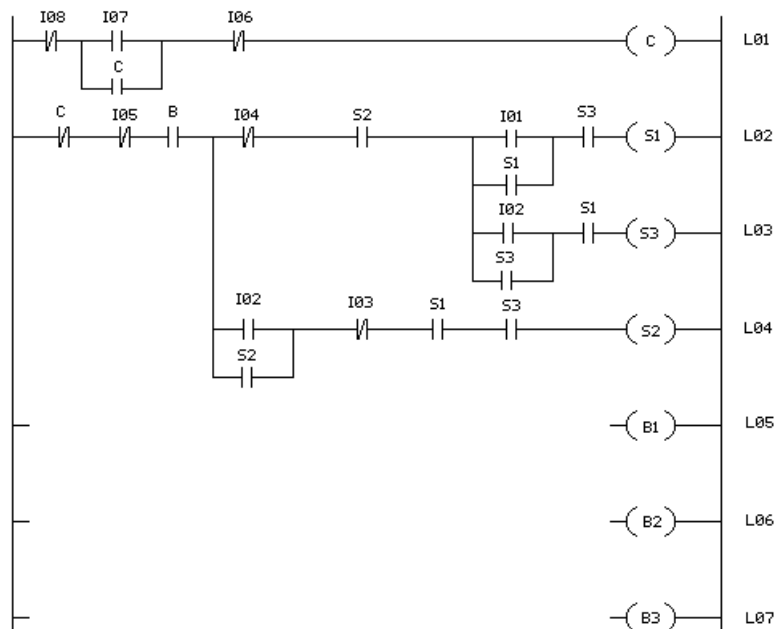
El cableado se debe codificar, para mantener una norma constructiva, y cada firma adopta la suya de acuerdo a criterios preestablecidos, pero una norma adecuada es titular el cable con la variable correspondiente al borne del módulo; además se puede codificar el color .

El elemento que se conecte al PLC, deberá disponer de un código mnemónico, con los cuales se debe confeccionar una lista de materiales especificando el código y marca comercial del elemento

En fase de proyecto siempre se debe considerar un 20% aprox. de entradas disponibles para eventuales ampliaciones del sistema, en el caso de las salidas puede ser más preciso debido que se conoce la cantidad de actuadores de antemano de un modo bastante poco variable.



A continuación se detalla un probable programa de PLC para el montacargas, existen características específicas de cada controlador que se deberán estudiar en cada caso particular.



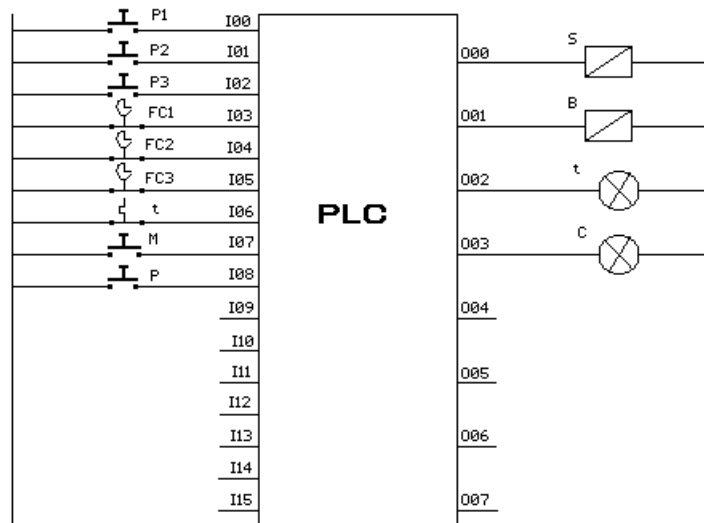
Obviamente que cualquier modificación en el sistema de control significará un cambio en el programa, pudiéndose realizar modificaciones sustanciales en las prestaciones del controlador.

El agregado de variables intermedias redonda solo en una mayor cantidad de líneas de programa sin significar un costo de modificación en las instalaciones.

Direccionamiento de las I/O

Cada variable conectada al PLC será direccionada en forma unívoca, a cada una le corresponde una dirección de la imagen del mapa de memoria.

Para dimensionar el mando será necesario conocer la cantidad de variables de entrada/salida y seleccionar una CPU capaz de direccionar esa cantidad de variables físicas. Las variables pueden ser de tipo local o remoto, dependiendo de la topología de la aplicación



Programación de PLC's

Introducción

Uno de los datos de la realidad en cuanto a la elaboración de software de PLC es que el fabricante / proyectista de máquinas / sistemas debe elegir entre los productos que le ofrecen una serie de fabricantes, muchas veces atendiendo las necesidades del cliente. En la mayoría de los casos se requieren equipos y software especiales para cada caso.

Existe una norma tendiente a la estandarización de los lenguajes de programación (IEC 1131-1). No obstante aún se está muy lejos de una plena portabilidad de software de PLC.

Existe bastante incertidumbre acerca de la posibilidad de que el PLC clásico sea reemplazado por la PC industrial, en lugar de esto los lenguajes clásicos como el

ladder, AWL, funcional serán perfeccionados y aún parcialmente sustituidos por otros lenguajes más elaborados y flexibles como 'structured text' y 'C'.

A éstas condiciones generales de entorno deben sumarse otras de diferentes y tan valdezas características como :

- *Exigencias del cliente respecto a la funcionalidad del sistema*
- *Incumplimiento de cronogramas y costos.*
- *Elevados costos de puesta en servicio, mantenimiento y debug.*
- *En muchos casos, la información se genera al final, en modo incompleto y desactualizada.*
- *Nuevas disposiciones y normas más exigentes.*
- *Soluciones particulares implicando mayores costos.*

En gran parte los inconvenientes enunciados se deben a la falta de un desarrollo sistemático del software. Como posible respuesta se sugiere algunas alternativas.

- *Procedimiento paralelo en la elaboración de programa PLC (a diferencia del procedimiento secuencial tradicional).*
- *Descripción de la función de control con objetos, en forma independiente de los equipos.*
- *Descripción de objetos de la función con gráficos de estado.*

Procedimientos paralelos en la programación del PLC

Un análisis de los procedimientos actuales para la construcción mecánica y aún eléctricas.

El trabajo constructivo en la parte mecánica implica un proceso de descripciones cada vez más detalladas. En el último nivel se generan planos y listas de unidades que contienen la secuencia constructiva y los detalles técnicos de los elementos básicos. Desde el punto de vista de una construcción eléctrica los elementos básicos son señalizadores y actuadores como por ejemplo motores y cilindros hidráulicas. En la elaboración de programas PLC se reconstruye la función de una instalación a partir de éstos documentos.

El método tiene algunas desventajas : los planos de construcción contienen demasiados detalles pero no hacen una descripción de las funciones ; la coordinación de los diferentes departamentos intervinientes se hace muy tarde y no se promueve la estandarización de las funciones de control.

El perfeccionamiento se logra por medio de etapas intermedias definidas en el proceso de optimización de la construcción que se guía por los diferentes niveles en la jerarquía de una planta. Cuando se llega a un nivel intermedio la correspondiente documentación se envía al correspondiente departamento. Además de la especificación de las unidades constructivas utilizadas , estos documentos comprenden una descripción de la función y de las interacciones. La construcción

mecánica y eléctrica (elaboración de programas PLC) trabajan en forma prácticamente paralela.

Ventajas de los diagramas basados en funciones

Este método de modelado ofrece las siguientes ventajas :

- Se trabaja independientemente del tipo de equipamiento.
- Las descripciones detalladas que se usan en la construcción mecánica se van transfiriendo a la construcción eléctrica como base para el desarrollo sistemático del software.
- Modularización y más posibilidades de reutilizar los objetos de la función.
- Posibilidad de diferentes variantes intercambiando objetos.

Descripción de las funciones de control con gráficos de estado

Los objetos de la función introducida en la técnica anterior representan funciones de control y deben ser descriptos formalmente. A los efectos de la descripción gráfica de las funciones se pueden emplear cadenas secuenciales, redes de Petri o gráficos de estado.

La elaboración de programas con gráficos de estado permiten una continuidad desde la descripción general de funciones hasta el código con el que se trabajará en la máquina.

La elaboración de programas con gráficos de estado requiere de una condición, y es contar con la debida modularización como la que se emplea en los diagramas detallados de las funciones.

Metodología en el uso de gráficos de estado

Condición: Modularización de la planta en la descripción más detallada.

Pasos a continuación :

- 1- La conducta de los objetos físicos de la función que tienen asignados elementos eléctricos básicos se describe como gráficos de elementos.
- 2- Se establecen gráficos secuenciales para coordinar los gráficos de elementos.
- 3- La representación de los gráficos de estado se traduce a un lenguaje de máquina por medio de un compilador.

Gráficos de elementos: cuando se programan los gráficos de elementos, en un primer paso se asignan los actuadores y sensores al objeto de la función. A continuación se analiza cuales son los posibles estados, se describirán en forma de círculos o rectángulos y se les da una denominación unívoca.

La estructura del gráfico de estado se fija determinando las transiciones entre los estados. Un elemento fundamental de un gráfico de estado es que el objeto

descripto ocupa en todo momento exactamente uno de los estados fijados , esto se aplica a los gráficos de elementos como a los gráficos secuenciales.

Gráficos secuenciales : Para coordinar los gráficos de elementos se recurre a diagramas secuenciales, ejemplo de la entrega de pallets). En general a cada objeto abstracto de la función corresponde un gráfico secuencial. Las instrucciones de estado en el gráfico secuencial son mayormente enviadas a objetos inferiores . Las transiciones se producen en función de los recibos generados por los objetos de la función que recibieron la instrucción. En tal sentido las órdenes y los recibos deben complementarse en los gráficos de elementos y ser unidos con los respectivos elementos en el gráfico secuencial. Los procesos paralelos son modelables enviando la orden a varios gráficos de elementos en forma simultánea.

Compiladores : Un compilador genera automáticamente a partir del gráfico de estado el código que los actuales PLC están en condiciones de interpretar. Del mismo modo es posible generar un código "C" lo que ilustra las ventajas que ofrece una descripción que no está basada en los equipos ya que existe la posibilidad de incorporar

- Control de diferentes fabricantes
- Hardware avanzado (PC industrial)

Análisis comparativo de los métodos descriptos

Los gráficos de estado son los que permiten la mejor representación que a diferencia de las cadenas organizativas pueden ordenarse como se desee y se necesita describir menos elementos que en el caso de las redes de Petri.

La experiencia indica que los gráficos de estado son apropiados como medios de comunicación entre la construcción mecánica y la eléctrica.

Debido que todas las informaciones que se necesitan en el modelado de un objeto de la función se encuentran encapsuladas en el gráfico, los gráficos de estado ofrecen las mejores posibilidades de reutilización y eventuales variaciones.

Los métodos formales para reconocer bloqueos de programa o zonas muertas, establecer/comprobar si algo es correcto o no es más avanzado en las redes de Petri.

Lenguajes de programación

En general los lenguajes pueden clasificarse según el nivel :

Lenguaje de máquina : es el de más bajo nivel y no es fácil de utilizar por programadores que no disponen de conocimientos profundos de electrónica digital, dado que se debe trabajar con el procesador y sus dispositivos periféricos. Se trabaja con los códigos de procesador y sus instrucciones básicas, y debe conocerse la arquitectura del procesador y de la CPU.

Lenguaje ensamblador : de nivel intermedio, incorporando al anterior estructuras de programación de alto nivel. Los programas fuente son traducidos a lenguaje de máquina por un programa ensamblador (assembler) proporcionado por el constructor.

Lenguajes evolucionados : son lenguajes de alto nivel que permiten abordar problemas de mando específicos y de alta complejidad, se asisten de compiladores. El programador puede ejecutar este tipo de programación sin conocer la arquitectura del hardware .