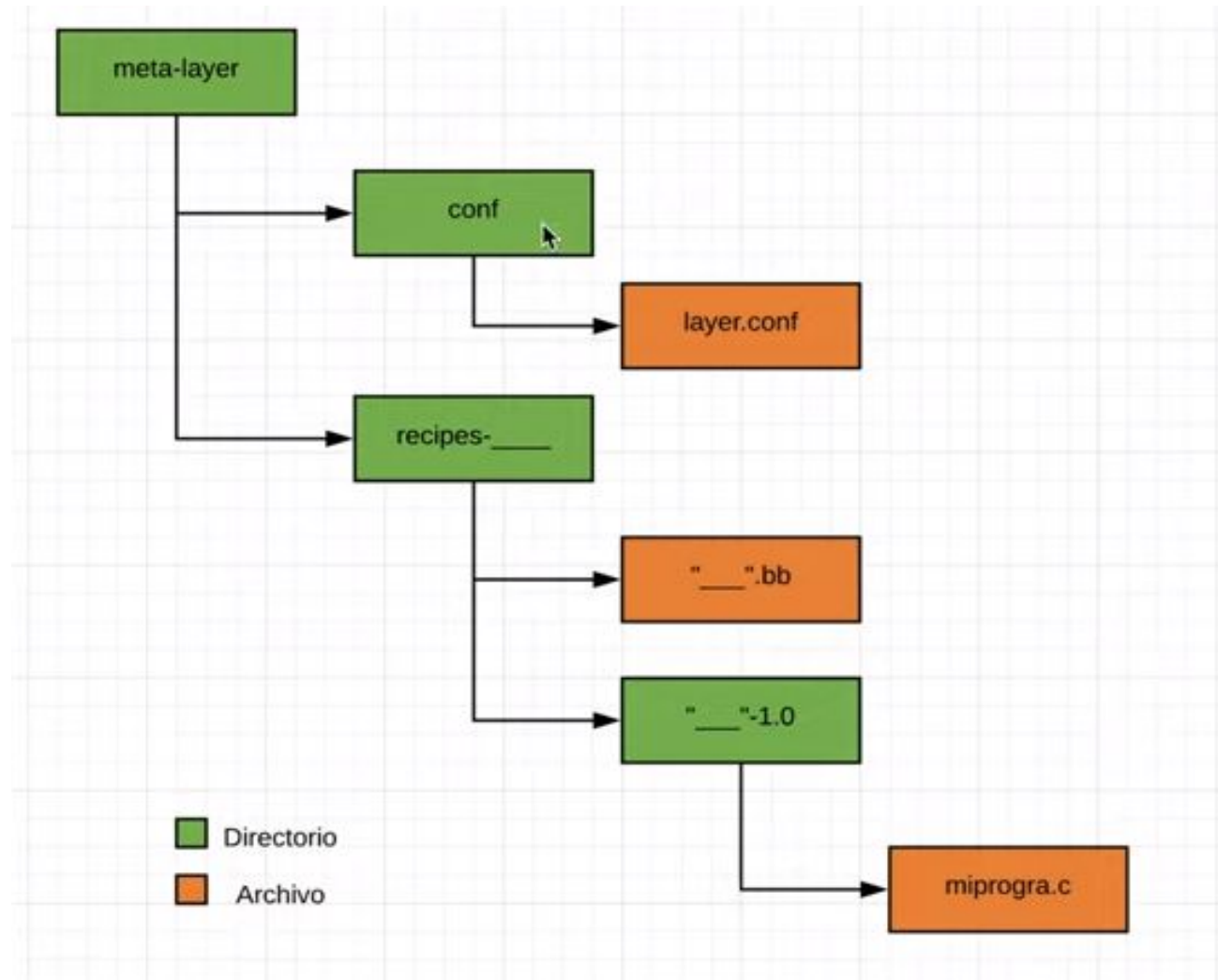


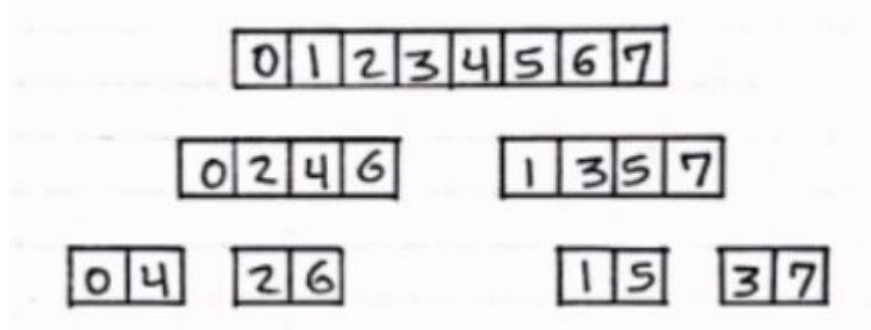
Diagrama básico para Yocto



Para ensamblador:

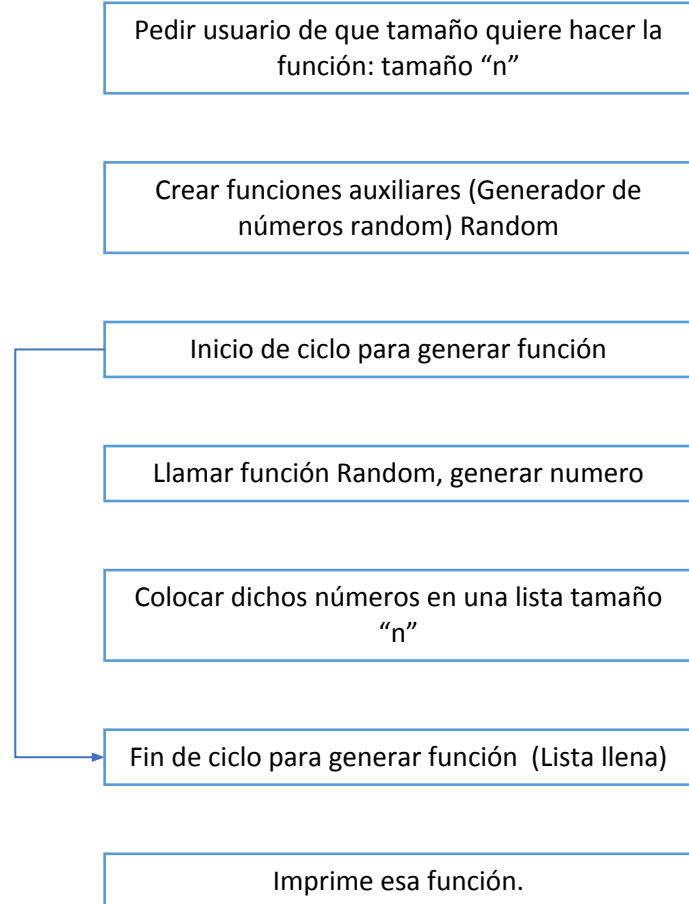
Se adjunta Nasm a
`meta/conf/bitbake.conf`

Para cálculo

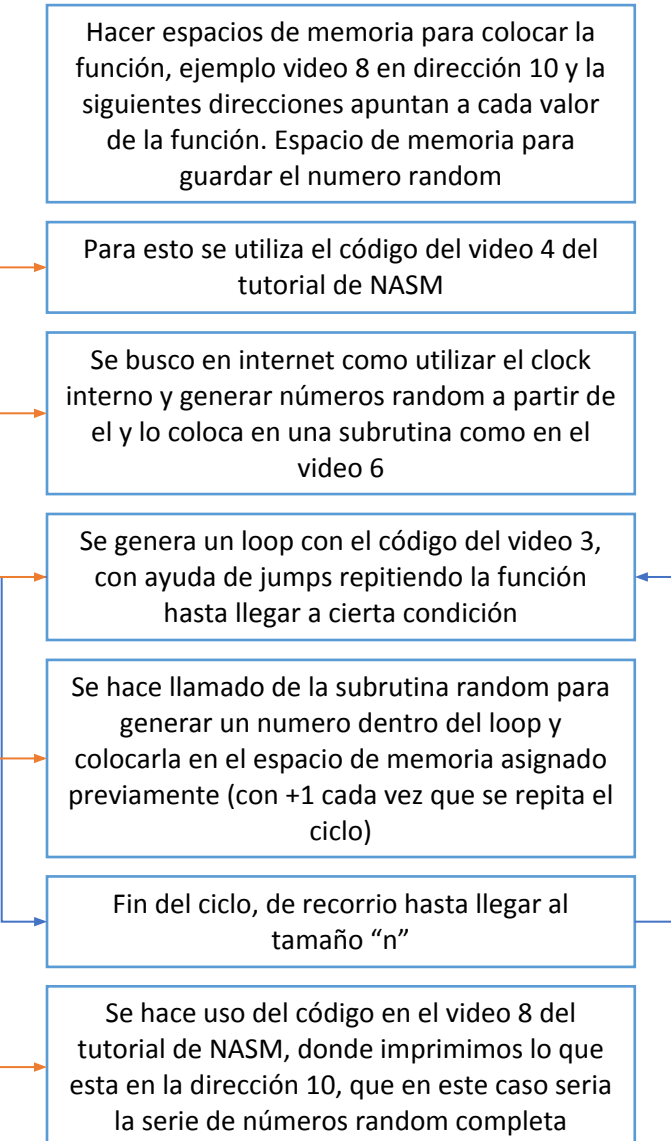


$$W_N^K = e^{-i \frac{2\pi}{N} k}$$

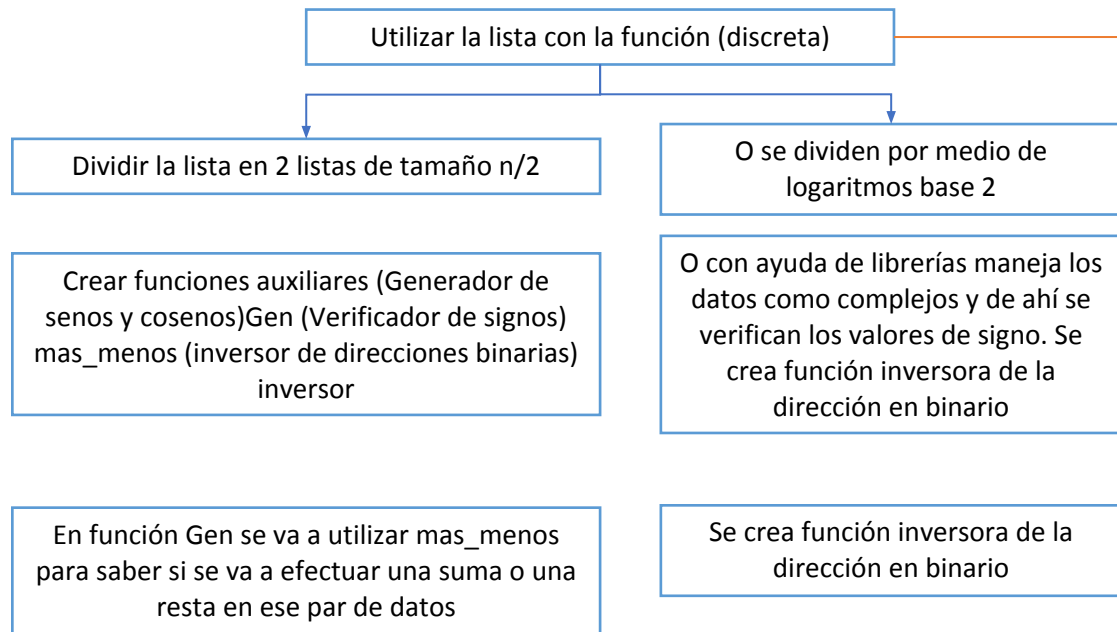
Primera parte (Generar función)



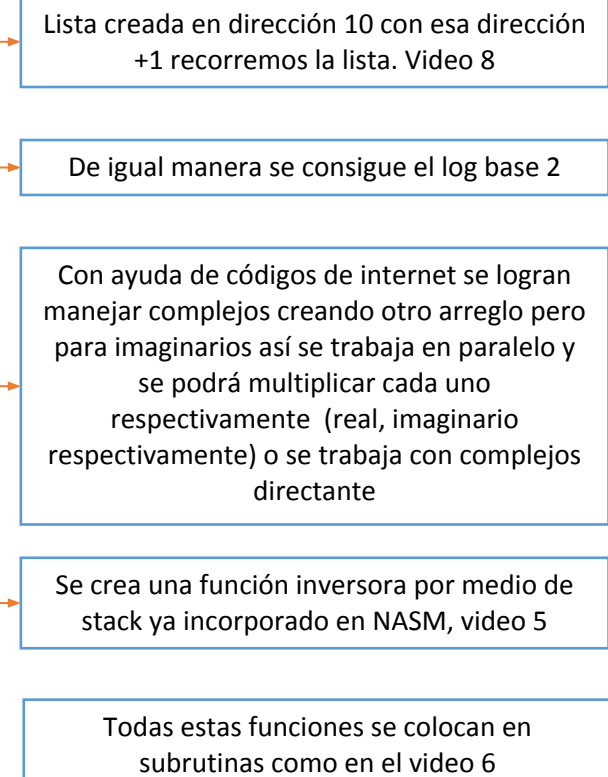
Primera parte (Generar función) NASM



Segunda parte (Preparar para hacer FFT)librerías



Segunda parte (Preparar para hacer FFT) NASM



Tercera parte (hacer FFT en función e imprimir resultado)

Inicio de ciclo para generar FFT

Llama el dato y utiliza "inversor" para generar su pareja y los guarda

Coloca el dato y su dato en dirección inversor en "Gen" y lo guarda

Este resultado lo multiplica por el primer dato y sobre escribe en la dirección del primer dato. Lo mismo ocurre con el dato invertido

Fin de ciclo para generar FFT (Utilizo toda la primera mitad de la lista)

Imprime resultado de la FFT sobre la función generada random.

Tercera parte (hacer FFT en función e imprimir resultado) NASM

Crea espacios de memorias para colocar inversor, multiplicación de la función y guardado de dirección de la función principal

Se genera un loop con el código del video 3, con ayuda de jumps repitiendo la función hasta llegar a cierta condición

Coloca dirección +1 para caer al primer dato y por medio de su dirección hace la subrutina inversión y encuentra su pareja y la guarda en inverso

Coloca el dato en la dirección que estamos trabajando y inverso en la función para efectuar la subrutina de senos y cosenos y lo guarda en multiplicación de función

Este ultimo dato lo guarda en la función principal y sigue el ciclo

Final del ciclo

Se hace uso del código en el video 8 del tutorial de NASM, donde imprimimos lo que esta en la dirección 10, que en este caso seria la función ya con la transformada de la FFT

Por complejidad primero se programo en C++ (programa ya conocido) y de ahí se paso a la lógica de NASM

Dificultades al implementar en NASM

- Números flotantes, se usa enteros
- Necesidad de una lista
- Saber el tamaño de las funciones y crear suficientes espacios de memoria
- Trabajar números complejos
- Trabajar senos y cosenos
- Trabajar logaritmos
- Manejo de registros para modificar una memoria

Soluciones al implementar en NASM

- Uso de compiladores para pasar código de C++/C a NASM
- Manejar la misma lógica de resolución del problema