¿Qué precisamos para modelar/implementar/ejecutar los fenómenos observables? **Representación de la información** , una forma de representar los datos con los que vamos a trabajar (números)

**Conservación de los datos** , un mecanismo que conserve el valor de nuestros datos, salvo que indiquemos lo contrario, a través del tiempo (memorias)

**Modelo de cómputo** , una definición de cómo operar con la información representada (set de instrucciones, microoperaciones)

¿Qué precisamos para modelar/implementar/ejecutar los fenómenos observables? **Operaciones aritmético/lógicas** , una implementación de operaciones aritmeticas y logicas basicas sobre nuestra información representada (ALU)

**Cómputo en base a valores previos** , una forma de resolver operaciones complejas a partir de ejecutar operaciones sencillas (atómicas) en una serie de pasos (circuitos secuenciales, microprogramación)

Para poder computar precisamos: **Representar** , **Conservar** , **Operar**.

Para poder computar precisamos: **Enteros acotados** , **Memorias** , **ALU**.

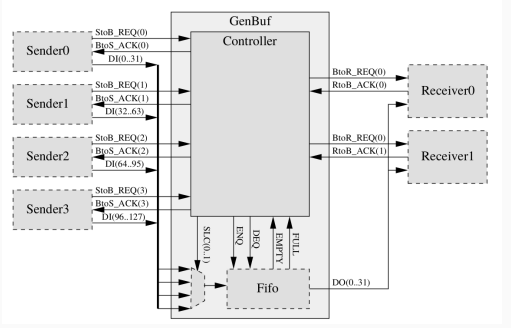
**Diseño e implementación de hardware**

Al igual que el software, la práctica del diseño y la implementación del hardware depende del siguiente grupo de actividades:

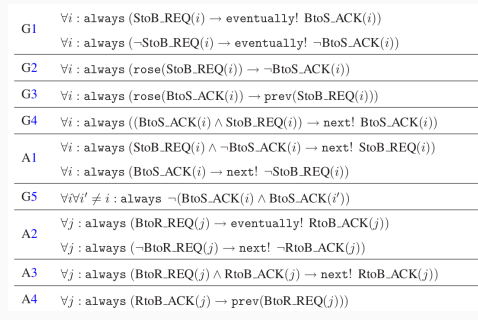
**Diseño → Especificación → Implementación → Validación → Verificación**

Del diseño a la implementación tenemos:

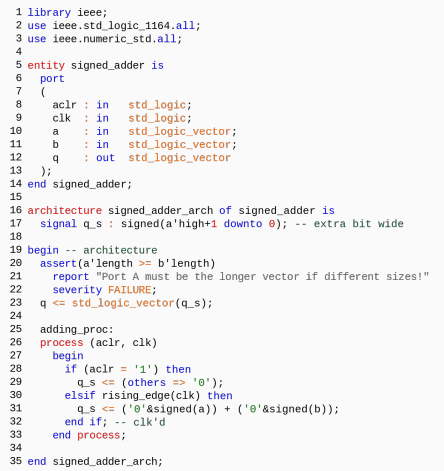
**Diseño** Descripción de arquitectura, división por vistas del sistema



**Especificación** Representación declarativa del comportamiento esperado

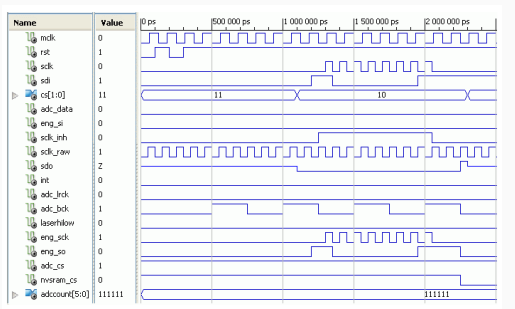


**Implementación** Interpretación imperativa de la especificación que permite producir una instancia operacional del sistema

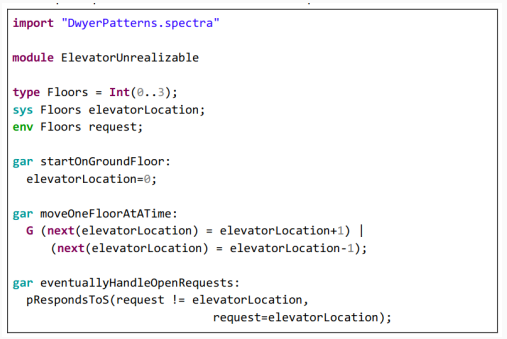


Y en las etapas post implementativas:

**Validación** Conjunto de pruebas no exhaustivas que prueban el comportamiento de la especificación y/o implementación.



**Verificación** Prueba de propiedades formales (con garantías basadas en algún mecanismo matemático) de la especificación y/o implementación.



**Conceptos principales**

**Razonamiento composicional** describe la práctica de dividir el funcionamiento del sistema en componentes que se encargan de una parte menor del comportamiento

**Abstracción** es el proceso a través del cual ocultamos detalles de un sistema o componente de acuerdo a las necesidades de la tarea u operación en curso

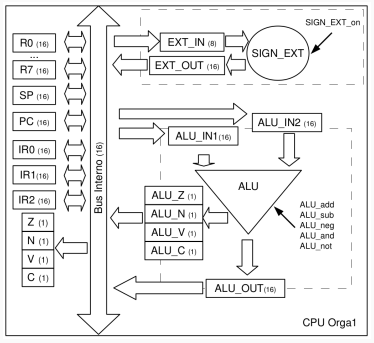
**Diseño por contratos** describe la forma en la que vinculamos componentes a partir de derechos de los que el componente goza y obligaciones que debe cumplir.

**Arquitectura del microprocesador**

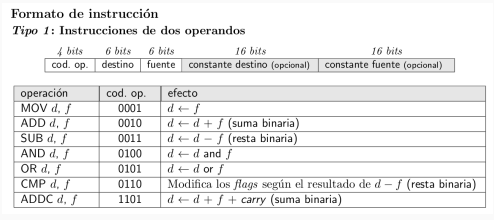
Recordemos que queremos diseñar e implementar un modelo de cómputo (microprocesador). Para el alcance de la materia vamos a interpretar a la arquitectura del mismo como:

**Diseño de sistema + Set de instrucciones → Microarquitectura**

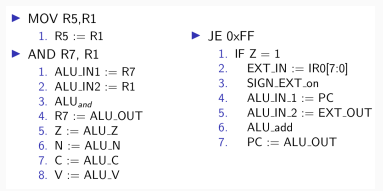
**Diseño de sistema** Elección de arquitectura y componentes involucrados.



**Set de instrucciones** Semántica de las capacidades de ejecución de nuestro microprocesador



**Microarquitectura** Implementación de nuestro set de instrucciones a través de la lógica de control de la arquitectura y componentes seleccionados.



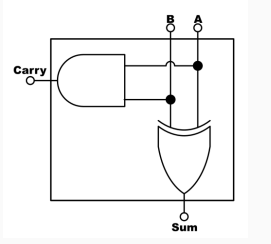
Los contenidos que vamos a presentar durante las clases prácticas son los siguientes:

**Lógica Combinatoria y Secuencial Diseño de un set de instrucciones**

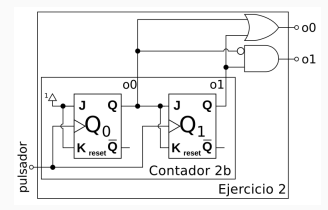
**Microprogramación**

**Manejo de memoria, Interrupciones, Caché Y Buses**

**Lógica Combinatoria** presenta los principios fundamentales para construir circuitos que implementen en un soporte electrónico la semántica de la lógica proposicional



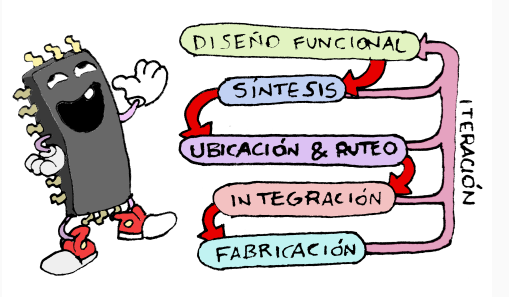
**Lógica Secuencial** introduce los elementos básicos para mantener el valor de un dato a lo largo del tiempo, los mecanismos de sincronización de circuitos y junto con estos la capacidad y técnicas que nos permiten descomponer e implementar un cómputo complejo en una secuencia de pasos atómicos.



**El diseño de un set de instrucciones** es el paso necesario para implementar nuestro microprocesador, donde vamos a definir el nombre y significado de las operaciones que deseamos poder ejecutar como parte de nuestros programas.

**La microprogramación** va describir la forma en que nuestros componentes sincrónicos interactúan para implementar las operaciones descritas en el set de instrucciones previamente definido.

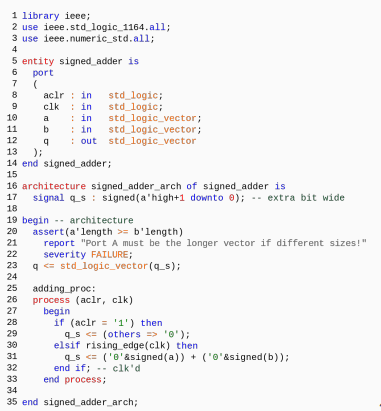
**Fabricación del hardware**



Hardware description language (HDL)

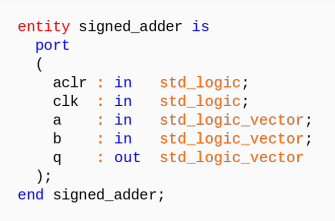
Del diseño a la implementación tenemos:

Un lenguaje de descripción de hardware, o HDL por sus siglas en inglés (hardware description language) es un lenguaje que describe la estructura y el comportamiento de un circuito digital. Los dos lenguajes más utilizados en la industria son VHDL y Verilog. En Orga 1 vamos a usar **VHDL**.



**Estructura de un documento VHDL**

Un documento VHDL va estar dividido en una descripción de interfaz y otra de comportamiento. La descripción de interfaz se define como **entity**, da un nombre al componente y enumera los tipos de señales expuestas, indicando su tamaño y si se trata de una entrada o una salida.



La descripción de comportamiento se define como architecture y define el vínculo funcional y/o lógico entre las señales de entrada, salida y cualquier elemento de representación interna, que se puede definir de forma estática o, por ej., en base al evento de flanco de reloj.

