System on Chip: diseño, plataformas y metodologías para desarrollarlos

Martín González Prieto, Rafael Ortegano,

Resumen—El siguiente informe se enfoca en el estudio de los System on Chip (SoC), su funcionamiento, sus aplicaciones y las perspectivas a futuro de su desarrollo. Se busca detallar cuáles son sus componentes básicas y comprender las ventajas y desventajas que tienen frente a sistemas tradicionales. Además, se proporciona una explicación y comparación de dos de las plataformas principales para su desarrollo, Soc-FPGA y ASIC. En lo que respecta al estado actual de ésta tecnología, se centra especialmente en las metodologías ágiles de desarrollo. Este análisis se lleva a cabo con el propósito de comprender la naturaleza de esta tecnología, evaluar su potencial para el avance tecnológico y abordar los desafíos que plantea.

Index Terms—IC, SoC, FPGA, ASIC, Agile

1. Introducción

En la actualidad estamos rodeados de circuitos integrados *(chips)* y cada día, estos se van complejizando con el fin de realizar más procesos. En la nueva era de la escalabilidad, las innovaciones en materiales y estructuras de dispositivos son tan importantes como el escalado dimensional. Anteriormente, la tendencia era utilizar transistores más pequeños para construir núcleos más grandes, que funcionen a mayor frecuencia y consuman más energía. La nueva era del escalado de microprocesadores es con un enfoque de *System on Chip (SoC)*, sistemas integrados en un solo chip que combinan un conjunto diverso de componentes. Estos componentes utilizan circuitos adaptativos, sensores integrados, técnicas sofisticadas de administración de energía y un mayor paralelismo para construir productos con múltiples núcleos y funciones. [1] En la actualidad, encuentran en todo tipo de dispositivos y sistemas, tanto embebidos como de propósito general.

Debido al gran avance de ésta tecnología, dispositivos que anteriormente eran específicos para un solo uso, como los teléfonos celulares, los cuales se utilizaban para realizar llamadas o enviar mensajes, ahora son dispositivos inteligentes y multipropósito. Entre estos nuevos propósitos se encuentra la creación y reproducción de contenidos multimedia, comunicación inalámbrica con otros dispositivos (Internet, Bluetooth), geolocalización, entre otros. Desde hace varios años, la utilización de sistemas integrados en un solo chip permitieron una gran innovación dentro de las prestaciones de los *smartphones* [5].

Un tema central al momento de diseñar un *System on Chip (SoC)*, es decidir la tecnología que se va a utilizar y el propósito del chip. Se puede optar por camino de los ASIC, que proveen un enfoque más específico, o implementaciones del estilo SoC-FPGA que nos brinden más versatibilidad a los cambios que puedan surgir sobre los requerimientos. Más adelante se desarrollará cuáles son las implicancias de optar por un camino u otro.

Por último, otro punto importante que debe ser tenido en cuenta para desarrollar un *System on Chip (SoC)* es la metodología, por esto, enfocaremos el estado del arte en esta área. Esto se debe a que con la masividad y alta demanda de este tipo de circuitos integrados, últimamente se buscan optimizar los procesos de producción y diseño para obtener soluciones que no sean obsoletas cuando lleguen al mercado.

2. Modelo básico de un SoC

Un sistema en chip o System on chip, es un circuito integrado que contiene en sí mismo todos los componentes necesarios para un sistema informático. Sin embargo, la complejidad del mismo yace en la capacidad de personalización que tiene, es decir, según su aplicación puntual, el sistema va a requerir unos componentes y una distribución en específico. Es por esto, que en la próxima sección, se desarrollará el diseño general de un SoC, sin entrar en componentes particulares, ya que estos dependerán del propósito del chip.

2.1. Componentes principales de un SoC

Básicamente, un sistema en chip consta de 3 tipos de componentes: procesadores, memorias e interconexiones [2].

2.1.1. Procesadores

Por lo general, siempre se cuenta con una unidad de procesamiento central (CPU) la cual determina la *Instruction set arquitecture* (ISA) e incluye registros y detalles de la *Aritmetic logic unit* (ALU).

Como muestra la *Figura 1*, se incluyen varios procesadores heterogéneos interconectados con una o más memorias. Usualmente, los SoC también tienen circuitos analógicos para manejar la información de sensores o la conversión de información analógica a digital (ADC), o también circuitos para conexiones inalámbricas. Para el caso de uso en

smartphones, se incluyen también componentes relacionados con el procesamiento de audio, el acceso a internet y funciones multimedia (vídeo, documentos, etc.). [2]

Cabe aclarar que con un diseño similar no siempre se puede encapsular todo en un solo chip y se suelen utilizar sistemas similares. Por ejemplo, se puede implementar en un módulo *System on module* (SoM), como Portenta X8 de Arduino, o en un *single-board computer* (SBC) como Raspberry Pi o un Arduino tradicional. Este tipo de aplicaciones son las que usualmente se ven en sistemas embebidos.

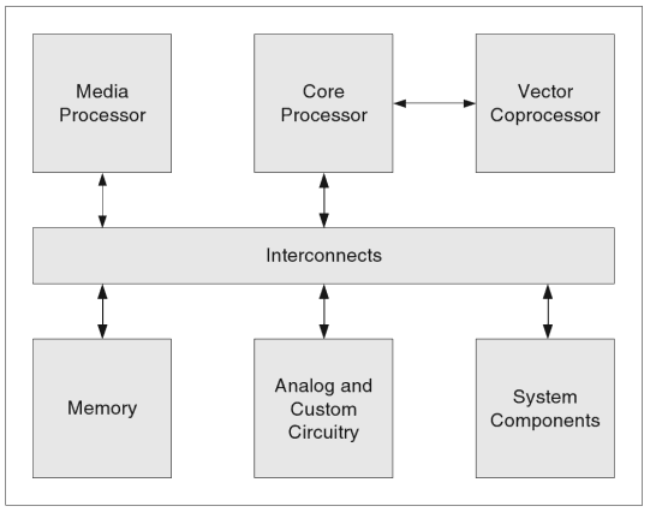

Figura 1. Diseño básico de un SoC [2]

2.1.2. GPU: Graphics Process Unit

La unidad de procesamiento de gráficos (GPU) es un procesador dedicado a procesamiento de gráficos. Este coprocesador se ha convertido en una parte integral de los principales sistemas informáticos actuales. A principios del siglo, ha habido un marcado aumento en el rendimiento y las capacidades de las GPU. La GPU moderna no es solo un potente motor gráfico, sino también un procesador programable altamente paralelo que presenta una aritmética máxima y un ancho de banda de memoria que supera sustancialmente a su contra parte de la CPU [7]. Un ejemplo conocido dentro de un SoC, es la serie Adreno dentro de procesadores Snapdragon de Qualcomm.

2.1.3. VPU: Vision Processing Unit

La unidad de procesamiento visual se especializa en cargas de trabajo exigentes de visión computarizada e IA. Al combinar un cómputo programable altamente paralelo con aceleración de IA por hardware, se logra un equilibrio de eficacia energética y desempeño. Últimamente, tiene una gran expansión en dispositivos de *Virtual reality* (VR) y en sensores para detección multimodal, como lo puede ser el reconocimiento facial en *smartphones* o detección de objetos [8].

2.1.4. DSP: Digital Signal Processor

El procesador de señales digitales (DSP) es un procesador dedicado al procesamiento de señales y representación de señales analógicas, haciendo uso también de un conversor analógico/digital para recibir las muestras. Este tipo de procesadores, se utiliza para el procesamiento de audio y vídeo en tiempo real, y en el último tiempo son omnipresentes en el mercado de *smartphones* y dispositivos de comunicaciones inalámbricas.

2.1.5. Memorias

Dentro de las memorias que dispone un SoC, es probable que existan memorias no volátiles (NVM), como los HDD, SSD o memorias Flash, pero nos vamos a centrar particularmente en las volátiles, aquellas indispensables para el funcionamiento y operatividad de los procesadores. La implementación más popular de este tipo de memorias son las RAM (*Random Access Memory)*, entre las cuales se encuentran las memorias estáticas (SRAM) y las memorias dinámicas (DRAM).

Las estáticas comúnmente se implementan en memorias caché de diferentes niveles (L1, L2, L3), las cuales, luego de los registros, serán las de próximo acceso para los procesadores.

Por otro lado, las dinámicas son mucho más baratas pero menos rápidas, y no se pueden integrar eficientemente en el chip con un procesador. Es por esto que siempre se encuentran por fuera como módulo de memoria. La implementación más popular de las DRAM son las aquellas que proveen información tanto en la subida como en la caída del flanco de clock, llamadas *Dual Data Rate* (DDR). [4]

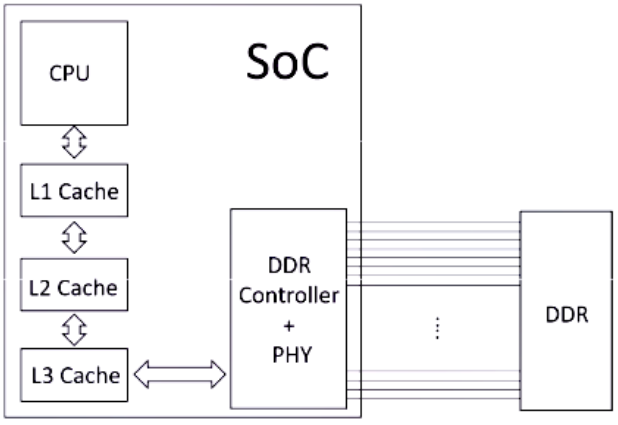

Figura 2. Tipo de memorias dentro de un SoC [4]

3. Ventajas y Desventajas con circuitos integrados tradicionales

3.1. Ventajas

- **Reducción de costos:**
  Al integrar varios componentes y por ende funcionalidades en un solo chip, los costos de fabricación se reducen (respecto a un sistema tradicional) y lo vuelve una solución económica para la producción a gran escala.
- **Versatilidad en diseño:**
  La posibilidad que brindan los SoC de customización es uno de los motivos principales que permiten su

desarrollo en dispositivos especializados, como lo pueden ser, los anteriormente nombrados, dispositivos IoT [9], dispositivos de uso médico, entre otros.

- **Fácilmente optimizables:**
  Es más fácil y simple centrarse en la optimización del rendimiento y las funcionalidades al estar todo integrado y no contar con problemas de interconectividad.
- **Rendimiento:**
  Con SoC altamente integrados, los núcleos se pueden utilizar para implementar funciones altamente computacionales en hardware.
- **Eficiencia energética:**
  Nuevamente, la simplificación del diseño y de los componentes, logra que haya un mejor rendimiento energético, un beneficio crucial para dispositivos con baterías como lo pueden ser los *smartphones*.
- **Portabilidad:**
  La capacidad de contener un sistema completo en un solo chip, brinda grandes ventajas para su implementación en dispositivos ligeros y pequeños.

3.2. Desventajas

- **Incremento de la complejidad:**
  Tanto el proceso de diseño como el de desarrollo puede llegar ser más intrincados de lo normal. Esta complejidad puede elevar los costos de desarrollo y es producto de la necesidad de integración.
- **Falta de modularidad:**
  No cuentan con la posibilidad de simplemente reemplazar un componente si este falla, se debe reemplazar todo el SOC. Así mismo, esto produce una pérdida de flexibilidad y adaptabilidad a nuevas exigencias.
- **Fallas de seguridad:**
  Con todos sus componentes integrados e interconectados, si hay una falla de seguridad en alguno de ellos, potencialmente pone en riesgo al resto.
- **Estrés térmico:**
  Al tener varios componentes y procesadores en un mismo circuito, hay un gran aumento en las densidades de temperaturas. Éstas altas temperaturas son un gran desafío ya que ponen en riesgo la fiabilidad[1] y pueden considerar un daño permanente del producto. Es por esto que se requiere un gran trabajo de optimización y de manejo de temperaturas para lograr mitigar este impacto. [6]

Las ventajas y desventajas indican claramente la razón por la cual la tecnología System-On-Chip es la más buscada por las industrias que se ocupan de la lógica integrada. Al integrar múltiples chips en un solo chip de silicio y producir un producto electrónico todo en uno, las empresas esperan obtener grandes beneficios en términos de fabricación, especialmente en mercados donde el precio y el tamaño del dispositivo son de importancia crítica. [3] [14]

1. La fiabilidad se refiere a la probabilidad de que un producto, sistema o servicio realice adecuadamente su función prevista durante un período de tiempo específico, o funcione en un entorno definido sin fallas.

4. Aplicaciones tecnológicas de los SoC

Los *System on Chip (SoC)* son muy comunes en los dispositivos actuales. Los diseñadores de hardware deben superar problemas complejos y desafiantes relacionados con el costo, el tiempo de comercialización, el rendimiento, la potencia, la capacidad, la calidad, entre otros.

La plataforma utilizada para implementar el diseño del SoC tiene un gran impacto en todas estas cuestiones y es una decisión fundamental que el diseñador de hardware debe tomar al inicio de cada nuevo proyecto. Debido a la complejidad y el rendimiento inherentes a la mayoría de las aplicaciones SoC, los diseñadores, históricamente, se limitaron a la tecnología ASIC [10]. En cambio, las tecnologías FPGA, caracterizadas por su capacidad de adaptación y generalidad de uso, ofrecen a los diseñadores de SoC otra plataforma de hardware viable para las aplicaciones actuales.

Es por esto que abordaremos estas dos opciones más detalladamente y a su vez, posteriormente compararemos su potencial en distintos puntos que son relevantes al momento de diseñar y desarrollar un *System on Chip (SoC)*.

4.1. Asic

Un *Application-Specific Integrated Circuit* (ASIC), o también llamado circuito integrado para aplicaciones específicas, es un circuito integrado (CI) diseñado a medida para una tarea o aplicación específica. A diferencia de las placas FPGA que pueden programarse para satisfacer una variedad de requisitos de casos de uso después de la fabricación, los diseños ASIC se adaptan en las etapas iniciales del proceso de diseño para abordar necesidades específicas.

Aunque el costo original del diseño puede ser alto, los ASIC pueden ser rentables para productos como teléfonos móviles y otros dispositivos de consumo populares que anticipan grandes volúmenes de producción. Un ASIC podría ser la mejor opción para diseñadores que buscan realizar funciones especializadas [12].

Es importante notar que si bien los ASIC ofrecen numerosas ventajas, no todas las empresas pueden aprovecharlos, ya que su desarrollo requiere una inversión financiera sustancial y conocimientos técnicos especializados. Como resultado de esto, el desarrollo de ASIC puede ser inaccesible para varias empresas debido a sus altos costos.

Debido a que las aplicaciones evolucionan mucho más rápido que la estandarización, que el tiempo de llegada al mercado y el costo de los SoCs específicos están aumentando constantemente, la configurabilidad posterior a la fabricación parece ser una necesidad. Sin embargo, los ASICs siguen contando con una gran importancia para ejecutar ciertas tareas muy específicas y con poca mutabilidad en el tiempo. [20]

4.2. Integración con módulos FPGA

Un *field programmable gate array* (FPGA) es un dispositivo semiconductor que contiene componentes lógicos programables e interconexiones programables entre ellos. Su naturaleza flexible y de propósito general provee ventajas:

- La reducción del tiempo para la salida al mercado de productos.

- La habilidad para ser reprogramadas después de haber salido al mercado a fin de corregir posibles errores.
- La reducción de los costos de investigación, diseño y prueba de un nuevo producto.

En resumen, uno de los puntos más fuertes de utilizar módulos FPGA es la flexibilidad que estos proveen [11].

Es por esto que una tendencia reciente ha sido llevar la arquitectura un paso más lejos, combinado los bloques lógicos e interconexiones de las tradicionales FPGAs, con microprocesadores y con periféricos relacionados para formar un SoC. Debido a que para segmentos de mercado altamente especializados y de bajo volumen, esto representa una alternativa viable a los SoCs [11].

Como resultado, tenemos los **SoC-FPGA (system on a chip field-programmable gate arrays)**, que son dispositivos que combinan en un chip un FPGA y un procesador de propósito general.

La alta capacidad de paralelismo de las FPGAs las ha situado como una opción rápida y económica, a medio camino entre los dispositivos de circuito integrado específico de la aplicación (ASIC) y los procesadores de propósito general. [13].

Una desventaja de las FPGAs es no poder contar con una interfaz de usuario simple e intuitiva como lo tienen los procesadores de propósito general, como así también su buen rendimiento para operaciones aritméticas básicas o el manejo de excepciones. Es por esto que al combinar estos dos elementos de procesamiento, los dispositivos SoC-FPGA se han convertido en una opción muy buena para el desarrollo de sistemas donde el procesamiento en tiempo real es clave. Utilizando estos dispositivos, la lógica dedicada en la FPGA puede ser empleada para resolver tareas que demandan baja latencia, dejando que el procesador se dedique a realizar tareas de alto nivel, como la interfaz de usuario [13]. Actualmente, hay dos fabricantes que concentran casi toda la oferta de sistemas SoC-FPGA. Estos son Xilinx e Intel.

4.3. Asic vs SoC-FPGA, ¿Qué plataforma elegir?

Exploraremos varios puntos a tener en cuenta cuando se quiere desarrollar un *System on Chip (SoC)*, en esta sección mencionaremos una serie de los mismos, detallando como esto impactará en el SoC dependiendo de que tecnología utilicemos.

- **Propósito:**
  Los FPGA se pueden reconfigurar con un diseño diferente. Incluso tienen la capacidad de reconfigurar una parte del chip mientras las áreas restantes del chip siguen funcionando. Esta característica se usa ampliamente en computación acelerada en centros de datos.
  Por otra parte, los ASICs son circuitos permanentes. Una vez que el circuito específico de la aplicación está grabado en silicio, no se puede cambiar. El circuito funcionará igual durante toda su vida útil.
  En resumen, si el *System on Chip (SoC)* que va a ser diseñado está destinado a una tarea específica, probablemente en este punto ASIC sea lo más adecuado, por el contrario, si necesitamos generalidad y capacidad de cambio, FPGA es una solución que se adapta mejor. [19]
- **Cantidad a producir:**
  Una característica fuerte de los ASIC, en contraposición de los FPGAs, es la gran capacidad que tienen para la producción a gran escala. Por lo tanto, son la mejor alternativa para productos de utilización masiva.
- **Costos:**
  Por un lado, los ASICs tienen una barrera de entrada muy alta en términos de costo, curva de aprendizaje, enlace con la fundición de semiconductores, etc. Iniciar el desarrollo de ASIC desde cero puede costar millones de dólares. Caso contrario FPGA, que se puede empezar a desarrollar con costos muy bajos. [19] A su vez, el alto costo por unidad de los FPGA de alta gama es prohibitivo para todas las aplicaciones, excepto para las de menor volumen (unos pocos miles de unidades por año) [10].
  Es relevante mencionar que para crear prototipos, validar diseños o conceptos, muchos prototipos ASIC se crean utilizando FPGA. Los propios fabricantes de procesadores importantes utilizan FPGA para validar sus System-on-Chips (SoC). Es más fácil asegurarse de que el diseño funcione correctamente según lo previsto mediante la creación de prototipos FPGA. En general, no se recomienda producir un prototipo de un diseño utilizando ASIC a menos que haya sido validado [19].
- **Uso de Energía:**
  Los FPGAs son menos eficientes energéticamente y requieren más potencia para la misma función que ASIC puede lograr con menor potencia. A su vez, los ASICs son mucho más eficiente energéticamente. El consumo de energía de los ASIC se puede controlar y optimizar. Dependiendo del *System on Chip (SoC)*, puede ser un punto importante a tener en cuenta.

Estos, si bien no son los únicos puntos, deben ser tenidos en cuenta al momento de diseñar un *System on Chip (SoC)*. En el caso de los FPGA, estos son adecuados para aplicaciones en las que requieren actualizaciones o posibles cambios, dónde el alto coste de los FPGA en producción de gran escala no es el factor decisivo. Por otra parte, los ASIC son adecuados para áreas de aplicación donde el diseño no necesita actualizaciones y se busca un uso específico.

Podemos concluir entonces que dependiendo del presupuesto que tengamos, el uso que vayamos a darle, la amplitud de mercado que necesitemos cubrir y el surgimiento de posibles actualizaciones/modificaciones son, entre otras cosas, las que nos marcan qué utilizar al momento de diseñar un *System on Chip (SoC)*.

5. METODOLOGÍAS DE DESARROLLO

Los diseños de System-on-Chip (SoC) ya no pertenecen a empresas especializadas como los fabricantes de procesadores. Empresas como Amazon, Apple, Tesla y Google han presentado sus chips desarrollados internamente. A su vez, empresas de menor escala y el mundo académico han iniciado diseños de SoC que a menudo se basan en RISC-V

y otros componentes de código abierto. Por lo tanto, los proyectos de SoC tienen alcance a una comunidad más grande e incluyen muchos desarrolladores nuevos que a menudo tienen más experiencia en ingeniería de software que en diseño de circuitos integrados de aplicaciones específicas (ASIC). Es por esto que hay más interés y experiencia en el desarrollo ágil que antes.

La diferencia clave con un proyecto de software es que el SoC es un producto físico. Incluye pasos de diseño específicos que requieren herramientas y conocimientos dedicados. El orden de los pasos es crítico y provoca largas cadenas de dependencia [15].

A su vez, en el desarrollo de un *System on Chip (SoC)*, los fallos son muy costosos en esfuerzo humano y tiempo, además de los costos de preparación del material de producción. Volver a ejecutar simulaciones, síntesis o diseño físico puede llevar semanas después de un cambio de diseño. En general, la situación desafía el enfoque tradicional de diseño de hardware que busca un desarrollo de SoC más rápido, más rentable y predecible, y los métodos ágiles se centran en resolver dichos desafíos *System on Chip (SoC)*.

En esta sección, veamos en brevedad dos de las principales metodologías de desarrollo utilizadas para los *System on Chip (SoC)*.

5.1. Model-Driven Development (MDD)

El desarrollo impulsado por modelos (MDD) es un enfoque top-down. El diseño comienza a partir de modelos abstractos y el flujo continúa a través de múltiples capas de modelado intermedias hasta la implementación de destino. Cada una de las capas de modelado agrega más detalles específicos de la implementación al modelo. MDD intenta una semántica de modelo bien definida, que permite prácticas de corrección por construcción y formalismo para la generación de código.

5.2. Unified Modeling Language (UML)

Se sabe que el Lenguaje Unificado de Modelado (UML) modela arquitecturas de software antes de su implementación. UML también se ha extendido al modelado de hardware. El estándar IP-XACT es un lenguaje de modelado para el desarrollo de SoC y existen herramientas de desarrollo IP-XACT como Kactus2.

6. Metodologías ágiles

Partiendo de las metodologías enunciadas en la sección anterior, hay todo un abanico de posibles prácticas ágiles para llevarlas a cabo y es donde principalmente queremos enfocar esta sección.

Scrum y Extreme Programming (XP) son las prácticas más conocidas para llevar a cabo proyectos de desarrollo de software. Scrum se centra en las prácticas de gestión de proyectos, y XP se centra principalmente en las directrices prácticas para el desarrollo de software, como el desarrollo basado en pruebas (TDD) y la integración continua (CI).

La adaptación de las prácticas de desarrollo ágil es común para diferentes dominios y tipos de proyectos. Ha habido un esfuerzo por establecer principios y prácticas para el desarrollo ágil de SoC por parte de UC Berkeley. Su propuesta incluye los siguientes cuatro principios:

- Prototipos fabricables incompletos sobre modelos con todas las funciones
- Equipos colaborativos y flexibles en lugar de rígidos.
- Mejora de herramientas y generadores sobre la mejora de la instancia.
- Respuesta al cambio siguiendo un plan (diseño iterativo).

Es considerado como el método ágil formalizado más conocido en el desarrollo de SoC debido a su definición bien documentada. A su vez, cuando se implementan enfoques ágiles para el desarrollo mecánico de *hardware*, se necesitan metodologías específicas. Sin embargo, en la actualidad, no existe gran variedad de estudios que se centren en el desarrollo ágil de hardware para los Soc.

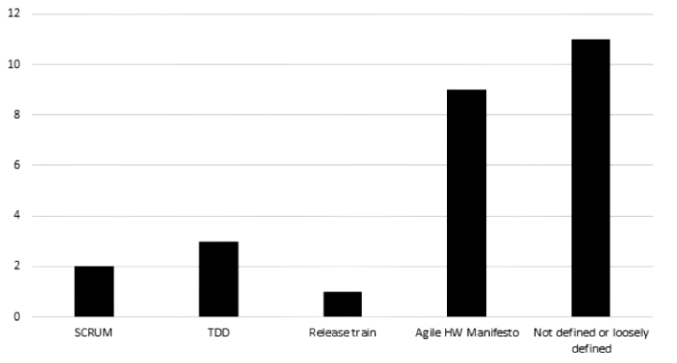

Figura 3. Aplicación práctica de metodologías ágiles de desarrollo [15]

Los resultados muestran claramente que los métodos y prácticas tradicionales de desarrollo ágil basados en *software* no se han aplicado en gran medida al desarrollo de *hardware*. Se desconoce el motivo y se necesita más investigación para estudiar esta observación. El desarrollo ágil de *hardware* no se ha definido rigurosamente y, por lo tanto, aún se encuentra en una fase muy temprana.

Está claro que en la mayoría de los casos los métodos o prácticas ágiles no están definidos y los métodos populares en ingeniería de software como SCRUM y XP no están presentes de manera significativa. Esto puede indicar que los desarrolladores de *hardware* no están familiarizados con los métodos de gestión de proyectos existentes o que no los consideran factibles. [15]

Hay muchos métodos y prácticas propuestos para aumentar la agilidad, pero ninguno de ellos es completamente novedoso o ágil como tal. La mayoría de los trabajos se centran en aumentar la eficiencia del desarrollo con prácticas MDD (abstracción, generación de código, transformaciones de modelos y síntesis), testing automático o también adaptando CI para ciertos procesos. [15] [16]

La investigación sobre el desarrollo ágil de *hardware* se encuentra todavía en una fase inicial, sin embargo, existen campos de investigación con gran potencial. Uno de ellos es ESP, una plataforma de investigación de código abierto para la realización de arquitecturas más escalables para SoC que integran componentes más heterogéneos, gracias a una metodología de diseño más flexible que se adapta

a diferentes lenguajes de especificación y flujos de diseño. [17] Otro ejemplo es UltraFast Design Methodology, una metodología creada por Xilinx que busca una realizar un proceso más rápido y eficiente minimizando el tamaño y la complejidad de los diseños destinados a FPGAs y SoCs. [18]

En resumen, el uso de métodos y prácticas ágiles conocidos en el dominio del hardware está limitado. Sin embargo, el número de publicaciones y la presencia en foros muestra un interés creciente en el desarrollo ágil de *hardware* SoC.

7. Conclusiones

A partir de lo explayado en este artículo, podemos llegar a varias conclusiones sobre la importancia de los *System on chip* en la actualidad y cuáles son sus desafíos a futuro. Primeramente, es una tecnología con un alto impacto en dispositivos de utilización masiva y de diferentes tipos de aplicación. Esto se da gracias a sus ventajas en performance (por su versatilidad en el diseño), en eficiencia energética complementada al uso de baterías y en su reducido tamaño. Por otro lado, las decisiones de diseño y desarrollo toman muchísima importancia producto de sus problemas para poder adaptarse a nuevos requerimientos posteriores a su fabricación, principalmente en el caso de los ASIC. Es en este campo dónde la investigación tiene un gran potencial. Si se encuentran metodologías ágiles de desarrollo que permitan abaratar los costos y tiempos de diseño y además permitan tomar decisiones óptimas en relación con el uso particular que tendrá el SoC, nos podemos encontrar con una tecnología increíblemente versátil y económica que acompañe la evolución de los cambios tecnológicos.

Referencias

[1] M. Bohr, *"The new era of scaling in an SoC world"* 2009 IEEE International Solid-State Circuits Conference - Digest of Technical Papers, San Francisco, CA, USA, 2009, pp. 23-28, doi: 10.1109/ISSCC.2009.4977293.

[2] M. J. Flynn, W. Luk, *Çomputer System Design: System-on-Chip"*. 2011. Available: https://books.google.com.pr/books?id=QXtyqHryAL4C

[3] N. Muralidharan, S. Wunnav and A. Noel,*"The System on Chip Technology"* Second LACCEI International Latin American and Caribbean Conference for Engineering and Technology (LACCEI'2004)l, 2004, Miami, Florida, USA, Available: https://www.laccei.org/LACCEI2004-Miami/papers

[4] Priyank Shukla,*"Types of Memories in Computing System-On-Chips"* Enero, 2018. Available: https://www.design-reuse.com/articles/43464/types-of-memories-in-computing-system-on-chips.html

[5] Yang, Chenyu. *"Vertical Structure and Innovation: A Study of the SoC and Smartphone Industries."* The RAND Journal of Economics 51, no. 3 (2020): 739–85. Available: http://www.jstor.org/stable/45380723.

[6] Coskun, Rosing, Mihic, De Micheli, Leblebici, *.Analysis and Optimization of MPSoC Reliability"*, Journal of Low Power Electronics, Volume 2, Number 1, April 2006, pp. 56-69(14) DOI: https://doi.org/10.1166/jolpe.2006.007

[7] J. D. Owens, M. Houston, D. Luebke, S. Green, J. E. Stone and J. C. Phillips, *"GPU Computing"* in Proceedings of the IEEE, vol. 96, no. 5, pp. 879-899, May 2008, doi: 10.1109/JPROC.2008.917757.

[8] Y. Cao, J. Bin, J. Hamari, E. Blasch, Z. Liu *"Multimodal Object Detection by Channel Switching and Spatial Attention"* Available: https://bit.ly/3tkV0Op

[9] Nolan, Stephen M. *"Power Management for Internet of Things (IoT) System on a Chip (SoC) Development"*. Design And Reuse. Retrieved September 25, 2018. Available: https://www.design-reuse.com/articles/42705/power-management-for-iot-soc-development.html

[10] Rick Mosher, *"Structured ASIC Based SoC Design"*2004, Dallas, TX USA. Available: https://www.design-reuse.com/articles/7058/structured-asic-based-soc-design.html

[11] José Manuel Marín de la Rosa, *"Field Programmable Gate Array (FPGA)"*. Available: https://bit.ly/3RRCl68

[12] ARM Glossary: ASIC

[13] Matías Javier Oliva, Pablo Andrés García, Enrique Mario Spinelli, Alejandro Luis Veiga *"SoC-FPGA systems for the acquisition and processing of electroencephalographic signals"* International Journal of Reconfigurable and Embedded Systems (IJRES) Vol. 10, No. 3, November 2021, pp. 237 248 ISSN: 2089-4864, DOI: 10.11591/ijres.v10.i3.pp237-248 237

[14] *.Advantages and Disadvantages of SoC (System on Chip)"* by Maven Silicon, September 8, 2023. Available: https://www.maven-silicon.com/blog/advantages-and-disadvantages-of-soc/

[15] Antti Rautakoura and Timo Hämäläinen. 2023. *"Does SoC Hardware Development Become Agile by Saying So: A Literature Review and Mapping Study"*. ACM Trans. Embed. Comput. Syst. 22, 3, Article 44 (May 2023), 27 pages. https://doi.org/10.1145/3578554

[16] Nicholas R. Pelland, *Çontinuous integration and delivery in hardware design"* 2021. Iowa State University. Available: https://dr.lib.iastate.edu/server/api/core/bitstreams/dfb28864-3e64-4003-a9fc-69f70253d464/content

[17] Paolo Mantovani, Davide Giri, Giuseppe Di Guglielmo, Luca Piccolboni, Joseph Zuckerman, Emilio G. Cota, Michele Petracca, Christian Pilato, and Luca P. Carloni *Agile SoC Development with Open ESP* Department of Computer Science, Columbia University in the City of New York, New York, NY 10027. Available: https://arxiv.org/pdf/2009.01178.pdf

[18] AMD Xilinx *UltraFast Design Methodology Guide for Xilinx FPGAs and SoCs*. Available: https://www.xilinx.com/support/documents/sw$_manuals/xilinx2022_1/ug949-vivado-design-methodology.pdf$

[19] Rohit Singh, *"FPGA Vs ASIC: Differences Between Them And Which One To Use?"* July, 2018 Available: https://numato.com/blog/differences-between-fpga-and-asics/

[20] Jari Nurmi, *"Processor Design: System-on-Chip Computing for ASICs and FPGAs"* Finland, Tampere University of Technology.Available: *http://www.stillart.ch/programming/Hardware/Processor$_Design_System-on-Chip_Comp_ASIC_FPGA.pdf$*