

Análisis Comparativo de Arquitecturas DSP y FPGA para el Control Táctico en Tiempo Real de la Unidad de Potencia Inteligente (IPU)

Resumen Ejecutivo

Este informe presenta un análisis exhaustivo de las arquitecturas de control digital para determinar la solución óptima para el controlador táctico en tiempo real de la Unidad de Potencia Inteligente (IPU). La IPU representa un avance significativo en los sistemas de energía fotovoltaica, pero su funcionalidad principal, el "Pistón Atómico" para la generación de potencia pulsada, introduce desafíos de control que superan las capacidades de los microcontroladores convencionales. Estos desafíos se centran en la necesidad de un procesamiento paralelo determinista y de latencia ultrabaja para gestionar simultáneamente múltiples bucles de control con diferentes criticidades temporales.

El análisis evalúa dos arquitecturas de control primarias: los Procesadores de Señales Digitales (DSP), optimizados para la computación secuencial de alta velocidad, y los Arrays de Puertas Programables en Campo (FPGA), que ofrecen un verdadero paralelismo a nivel de hardware. La evaluación se extiende a las soluciones de Sistema en un Chip (SoC), que integran un núcleo de procesador con tejido de FPGA.

La conclusión central de este informe es que una **arquitectura de SoC, que combina un sistema de procesamiento (PS) con lógica programable (PL), representa la solución más eficiente, modular y rentable para el controlador táctico de la IPU**. La lógica programable de la FPGA es ideal para implementar los bucles de control de potencia pulsada de tiempo real estricto, garantizando un rendimiento determinista y de latencia mínima. Simultáneamente, el sistema de procesamiento basado en ARM ofrece la flexibilidad necesaria para gestionar algoritmos complejos de nivel superior, como los Sistemas de Gestión de Baterías (BMS), el Seguimiento del Punto de Máxima Potencia (MPPT), las comunicaciones y futuras implementaciones de inteligencia artificial. Este enfoque híbrido mitiga los riesgos de desarrollo y rendimiento, proporcionando una plataforma robusta y preparada para el futuro para la familia de productos IPU.

Sección 1: El Desafío del Control en Tiempo Real de la IPU

Esta sección establece el contexto técnico definiendo la IPU y sus demandas operativas únicas. Se argumenta que la IPU no es un convertidor de potencia estándar, sino un sistema complejo y de alto rendimiento cuyos requisitos superan las capacidades de los microcontroladores comunes, necesitando un controlador táctico en tiempo real dedicado.

1.1. Definición de la Unidad de Potencia Inteligente (IPU) y el Concepto "Pistón

Atómico"

La Unidad de Potencia Inteligente (IPU) es un sistema avanzado diseñado para optimizar la gestión de la energía en sistemas fotovoltaicos. Integra de forma sinérgica la generación fotovoltaica, el almacenamiento en baterías y el almacenamiento de alta potencia en supercondensadores.[1, 1] El núcleo de su innovación reside en el concepto de "Pistón Atómico", un módulo de potencia pulsada que combina la descarga capacitiva con el retroceso inductivo para generar ráfagas de potencia instantáneas y de alta intensidad.[1]

Este proceso de generación de pulsos se basa en la conmutación de alta potencia y alta frecuencia de componentes semiconductores como MOSFETs e IGBTs. Esta conmutación, esencial para la funcionalidad del "Pistón Atómico", genera picos de calor significativos y pulsados que deben ser gestionados para no degradar los componentes ni reducir la eficiencia del sistema.[1, 1] Además, el sistema debe gestionar un flujo de potencia bidireccional complejo entre todos sus elementos: el arreglo fotovoltaico, las baterías, los supercondensadores y la carga final.[1, 1] En esencia, el "Pistón Atómico" transforma la IPU de un simple sistema de conversión de energía a un sistema de modelado de energía de precisión, donde el control temporal y la orquestación de la transferencia de energía son primordiales.

1.2. Deconstrucción de los Requisitos del Control Táctico: Latencia, Paralelismo y Determinismo

La naturaleza de la IPU, y en particular de su "Pistón Atómico", impone requisitos de control que van más allá de la simple regulación. El sistema exige un control altamente sofisticado para la transferencia rápida de energía y una temporización precisa para la generación de potencia pulsada.[1] Esto se traduce en la necesidad de manejar bucles de retroalimentación de alta velocidad y predecir estados de conmutación óptimos para maximizar la eficiencia.[1] Aunque la frecuencia de conmutación base de la electrónica de potencia se especifica en 10 kHz [1], las acciones de control para la generación de pulsos deben ocurrir en escalas de tiempo mucho más rápidas, del orden de microsegundos o incluso nanosegundos.

El término "controlador táctico" es, por lo tanto, apropiado. Debe tomar decisiones a nivel de microsegundos para la generación de pulsos mientras gestiona concurrentemente tareas en la escala de milisegundos para el BMS y el MPPT. Esto introduce tres requisitos críticos e interrelacionados:

- **Latencia ultrabaja:** El tiempo que transcurre desde la medición de un sensor (por ejemplo, corriente o voltaje) hasta la respuesta del actuador (por ejemplo, la actualización de una señal de modulación por ancho de pulso, o PWM) debe ser mínimo y predecible. Esto es fundamental para controlar con precisión los eventos de conmutación de alta velocidad del "Pistón Atómico".
- **Paralelismo real:** Múltiples bucles de control, cada uno con su propia escala de tiempo y criticidad, deben ejecutarse de forma simultánea y sin competir por los recursos de procesamiento. El control de la potencia pulsada no puede ser retrasado o afectado por la ejecución de los algoritmos del BMS o MPPT.
- **Determinismo:** El tiempo de ejecución de cada bucle de control debe ser predecible y

consistente, sin variaciones o "jitter". Esto es esencial para garantizar un funcionamiento estable y fiable, especialmente para la conmutación de potencia, que es crítica para la seguridad del sistema.

1.3. Subsistemas de Control Críticos: MPPT, BMS y Gestión de Potencia Pulsada de Alta Frecuencia

El controlador táctico de la IPU debe orquestar de forma concurrente tres subsistemas de control fundamentales:

- **Seguimiento del Punto de Máxima Potencia (MPPT):** Para maximizar la extracción de energía del arreglo fotovoltaico bajo condiciones climáticas variables, es necesario implementar algoritmos de MPPT. Estos pueden ir desde los métodos tradicionales como Perturbación y Observación (P&O) y Conductancia Incremental (Inc) [1, 2, 3, 4] hasta enfoques más avanzados basados en Lógica Difusa o Redes Neuronales que ofrecen un mejor rendimiento en condiciones cambiantes.[5, 6, 7, 8, 9, 10]
- **Sistema de Gestión de Baterías (BMS):** Este subsistema es esencial para monitorizar y controlar los parámetros críticos de la batería, como el voltaje, la corriente, la temperatura, el Estado de Carga (SOC) y el Estado de Salud (SOH). Un BMS robusto optimiza el rendimiento y la vida útil de la batería, garantizando su operación segura.[1, 11, 12, 13, 14, 15, 16, 17]
- **Control de Potencia Pulsada:** Este es el corazón funcional del "Pistón Atómico". Requiere un control preciso y de muy alta frecuencia sobre los interruptores de potencia (MOSFETs/IGBTs) para gestionar la carga rápida de los condensadores y el control del retroceso inductivo, que es lo que genera el pulso de alta potencia.[1, 1, 1] Este es un requisito de tiempo real estricto (hard real-time).

Estos tres subsistemas operan en dominios de control distintos con diferentes constantes de tiempo y niveles de criticidad. Un controlador táctico exitoso debe ser arquitectónicamente capaz de particionar estas tareas para asegurar que el control de pulsos, que es de tiempo real estricto, nunca se vea comprometido por la ejecución de las funciones de MPPT y BMS, que son de tiempo real más flexible (soft real-time). La gestión de estas tareas concurrentes y heterogéneas es el desafío central que define la elección de la arquitectura de control. Un procesador secuencial, como un DSP, gestiona la concurrencia mediante interrupciones y priorización de tareas, lo que puede llevar a conflictos: una interrupción de alta frecuencia para el control de pulsos podría privar de tiempo de cómputo a los algoritmos de MPPT y BMS, degradando su rendimiento. Por el contrario, si una tarea de menor prioridad deshabilita las interrupciones durante un período prolongado, podría introducir una latencia inaceptable en la generación de pulsos. Esta tensión inherente favorece una arquitectura que sobresalga en el paralelismo verdadero y determinista, donde cada bucle de control pueda operar de forma independiente sin interferencias.

Sección 2: Arquitecturas Candidatas para el Control en Tiempo Real

Esta sección ofrece un análisis arquitectónico profundo de los DSP y las FPGA, yendo más allá de las descripciones de alto nivel para analizar las características específicas relevantes para el desafío de control de la electrónica de potencia de la IPU. También se introduce el SoC

como una potente solución híbrida.

2.1. El Procesador de Señales Digitales (DSP): Una Arquitectura para la Computación Secuencial de Alta Velocidad

Un Procesador de Señales Digitales (DSP) es un microprocesador especializado cuya arquitectura está optimizada para las necesidades matemáticas del procesamiento de señales digitales.[18] Su diseño se centra en ejecutar algoritmos complejos de forma secuencial y a muy alta velocidad. Las características clave de su arquitectura incluyen:

- **Motor de Cómputo de Alta Velocidad:** Los DSP están diseñados para realizar funciones matemáticas como suma, resta y multiplicación de forma extremadamente rápida.[19] Esto se logra a través de unidades de hardware dedicadas, como la unidad de Multiplicación-Acumulación (MAC), que es fundamental para los filtros digitales y los algoritmos de control.[18, 20]
- **Arquitectura de Memoria Optimizada:** Para evitar cuellos de botella, los DSP suelen emplear arquitecturas de memoria avanzadas como la Harvard o la Harvard Modificada. Estas arquitecturas utilizan buses de datos y de programa separados, lo que permite al procesador obtener una nueva instrucción y datos de forma simultánea en un solo ciclo de reloj, aumentando significativamente el rendimiento.[18, 19]
- **Arquitectura Pipelined:** La segmentación de instrucciones (pipelining) permite que el DSP solape las fases de búsqueda, decodificación y ejecución de múltiples instrucciones, lo que resulta en un mayor rendimiento computacional.[18, 20]
- **Periféricos Especializados:** Los DSP diseñados para electrónica de potencia, como la serie C2000 de Texas Instruments, integran periféricos cruciales. Estos incluyen PWM de alta resolución (HRPWM) para un control preciso de los interruptores de potencia, convertidores analógico-digitales (ADC) rápidos y de alta precisión para la adquisición de señales, y comparadores optimizados.[21, 22, 23, 1, 24] Algunos modelos, como los de la serie C2000, también incorporan coprocesadores como el Acelerador de Ley de Control (CLA) para descargar tareas de control de la CPU principal y reducir la latencia.[22, 24]

En el contexto de la IPU, la arquitectura de un DSP es excelente para ejecutar de manera eficiente un único algoritmo de control complejo, como podría ser un avanzado algoritmo de MPPT o un filtro de Kalman para la estimación del estado de la batería. Sin embargo, su naturaleza fundamentalmente secuencial representa una limitación. La gestión de tareas verdaderamente concurrentes y con temporización determinista se basa en un sistema de interrupciones y prioridades. Aunque eficaz, este modelo puede introducir latencia y jitter, especialmente cuando una tarea de alta frecuencia y alta prioridad (como el control de pulsos) debe coexistir con otras tareas computacionalmente intensivas.

2.2. El Array de Puertas Programables en Campo (FPGA): Una Arquitectura para el Procesamiento Paralelo Real

Un Array de Puertas Programables en Campo (FPGA) es un circuito integrado que consiste en una matriz de bloques lógicos reconfigurables que pueden ser programados para crear una implementación de hardware personalizada de una aplicación.[25] A diferencia de los

procesadores, las FPGA no tienen un sistema operativo y utilizan hardware dedicado para cada operación lógica, lo que permite un paralelismo real donde diferentes procesos no compiten por los mismos recursos.[26] Sus componentes principales son:

- **Bloques Lógicos Configurables (CLB) y Tablas de Búsqueda (LUT):** Son las unidades programables fundamentales. Una LUT puede ser configurada para implementar cualquier función lógica booleana, actuando como una tabla de verdad personalizable.[26]
- **Interconexiones Programables:** Una densa red de enrutamiento que conecta los CLB entre sí, permitiendo la creación de rutas de datos personalizadas para una aplicación específica.[27]
- **Bloques de IP (Propiedad Intelectual) de Hardware Dedicado:** Las FPGA modernas no solo contienen lógica programable, sino también bloques de hardware optimizados para funciones comunes. Los más relevantes para el control de potencia son:
 - **Bloques de RAM (BRAM):** Memoria en el chip para un almacenamiento y acceso a datos de alta velocidad.[26, 27]
 - **Slices de DSP:** Unidades de hardware dedicadas que realizan operaciones matemáticas de alta velocidad, como multiplicaciones y acumulaciones, de manera mucho más eficiente que si se implementaran en la lógica programable general.[26, 27, 28]

La arquitectura de una FPGA es, en efecto, un "lienzo en blanco" de recursos de hardware paralelos. Para la IPU, esto significa que un diseñador puede crear tres controladores de hardware *separados e independientes* dentro del mismo chip: uno para el pulso del "Pistón Atómico", otro para el MPPT y un tercero para el BMS. Cada uno de estos controladores tendría su propia lógica, temporizadores y máquinas de estado dedicados, funcionando en paralelo con una precisión de ciclo de reloj. Esto proporciona un rendimiento determinista y de latencia ultrabaja para el bucle de control de pulso crítico, independientemente de lo que los otros bucles estén haciendo, eliminando por completo el conflicto de programación inherente a un sistema secuencial.

2.3. El Sistema en un Chip (SoC): Una Arquitectura Híbrida para el Control y la Gestión Integrados

Un Sistema en un Chip (SoC), como la familia Zynq-7000 de AMD/Xilinx, representa la fusión de las dos arquitecturas anteriores. Integra en un único encapsulado un **Sistema de Procesamiento (PS)**, que típicamente contiene núcleos de procesador ARM (como los Cortex-A9), y **Lógica Programable (PL)**, que es tejido de FPGA.[29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42] Ambos dominios, el PS y la PL, están estrechamente acoplados mediante miles de interconexiones de alta velocidad, lo que permite una comunicación de gran ancho de banda y baja latencia entre ellos.[38, 39, 40]

Esta arquitectura híbrida ofrece una solución ideal para el controlador táctico de la IPU al permitir una partición inteligente de las tareas, asignando cada una al recurso más adecuado:

- **Lógica Programable (PL) para el Control de Tiempo Real Estricto:** Las tareas que exigen la máxima velocidad y un determinismo absoluto, como la generación de pulsos del "Pistón Atómico" y los bucles de control PWM de los convertidores de potencia, se

implementan directamente en el tejido de la FPGA. Al ser realizadas en hardware dedicado, se garantiza la latencia más baja posible y una temporización sin jitter, inmune a las interrupciones del software.

- **Sistema de Procesamiento (PS) para la Gestión del Sistema y la Flexibilidad:** Las tareas menos críticas en cuanto a la temporización pero más complejas desde el punto de vista algorítmico se ejecutan en los núcleos ARM. Esto incluye la lógica de alto nivel del BMS (que puede implicar algoritmos complejos de estimación del estado de salud), estrategias avanzadas de MPPT (especialmente las basadas en aprendizaje automático), la gestión de protocolos de comunicación (como CAN bus o Ethernet), la ejecución de un sistema operativo (como Linux o un RTOS) y la provisión de interfaces de usuario. Este enfoque basado en software proporciona una enorme flexibilidad para futuras actualizaciones de algoritmos y la adición de nuevas funcionalidades sin necesidad de re-sintetizar el hardware.

La elección entre un DSP y una FPGA no es una mera preferencia, sino una decisión arquitectónica fundamental que define toda la filosofía de control del sistema. Un enfoque basado en DSP centraliza la inteligencia en un modelo secuencial impulsado por software, donde el principal desafío de diseño es la gestión del tiempo real a través de la programación de interrupciones y tareas. Un enfoque basado en FPGA descentraliza la inteligencia en bloques de hardware paralelos e independientes, donde el principal desafío es el diseño en lenguaje de descripción de hardware (HDL) y la asignación de recursos. La arquitectura SoC permite al diseñador evitar los compromisos de forzar una tarea de tiempo real estricto en un paradigma de software o un algoritmo complejo y en evolución en una implementación de hardware rígida. Al permitir que cada subproblema se resuelva con la herramienta más adecuada (hardware para la velocidad, software para la flexibilidad) en un solo chip, la arquitectura SoC se alinea directamente con el problema de control heterogéneo de la IPU, reduciendo tanto el riesgo de desarrollo como el de rendimiento.

Sección 3: Análisis de Rendimiento y Eficiencia

Esta sección realiza una comparación cuantitativa y cualitativa de las arquitecturas candidatas, centrándose en las métricas más críticas para la IPU: latencia, rendimiento y eficiencia energética.

3.1. Latencia y Rendimiento: Comparación de la Velocidad de Ejecución del Bucle de Control

La capacidad de un controlador para responder rápidamente a los cambios del sistema es fundamental en la electrónica de potencia, y es aquí donde las diferencias arquitectónicas entre DSP y FPGA se manifiestan de forma más clara.

- **DSP:** Los DSP de control en tiempo real, como la serie C2000 de TI, están altamente optimizados para una baja latencia. Incorporan características como rutas de señal de latencia cero desde el ADC hasta el PWM y aceleradores dedicados como el CLA que operan independientemente de la CPU principal.[22, 24] Un C2000 a 200 MHz puede ofrecer un rendimiento en bucles de control típico equivalente al de un procesador Cortex-M7 a 280 MHz.[22] Sin embargo, su naturaleza secuencial sigue siendo una

limitación inherente. La ejecución de algoritmos complejos o la gestión de múltiples tareas a través de interrupciones y un planificador introduce inevitablemente una sobrecarga y una variabilidad en la latencia (jitter).[43, 44]

- **FPGA:** Las FPGA logran una latencia extremadamente baja y un alto rendimiento gracias a su paralelismo de hardware real.[26] Los bucles de control pueden implementarse con una precisión a nivel de nanosegundos y una temporización completamente determinista, ya que no hay sobrecarga de un sistema operativo o un planificador de tareas.[27, 25] Esto permite que múltiples bucles de control se ejecuten a velocidades de cientos de kilohercios de forma simultánea y sin interferencias.[25] Para aplicaciones que requieren la monitorización de muchas señales en paralelo, como es el caso de la IPU, las FPGA son intrínsecamente superiores.[45]
- **SoC (Zynq):** Esta arquitectura combina lo mejor de ambos mundos. La lógica programable (PL) ofrece el procesamiento paralelo de baja latencia de la FPGA para los bucles críticos, mientras que los núcleos ARM (PS) se encargan del procesamiento de alto nivel.[35, 40] El acoplamiento estrecho entre el PS y la PL, a través de miles de interconexiones en el chip, proporciona un ancho de banda masivo y una comunicación de baja latencia entre el hardware y el software.[39, 40]

Para la funcionalidad del "Pistón Atómico" de la IPU, la latencia determinista a nivel de hardware de la FPGA es una ventaja decisiva. Mientras que un DSP de gama alta puede lograr tiempos de bucle muy rápidos para una *única* tarea, una FPGA puede garantizar la temporización del bucle de pulso *mientras procesa simultáneamente* los bucles de MPPT y BMS. Un DSP tendría que atender el pulso a través de una interrupción de alta prioridad, lo que aún podría tener jitter y necesariamente detendría la ejecución de otras tareas esenciales. La arquitectura FPGA elimina esta contienda por completo, asignando hardware dedicado a cada tarea.

3.2. Eficiencia Energética: Una Mirada Matizada al Consumo vs. Trabajo por Vatio

La eficiencia energética es un factor crítico, pero debe evaluarse no solo en términos de consumo de energía absoluto, sino como rendimiento por vatio.

- **DSP:** Generalmente, se considera que los DSP son más eficientes energéticamente para tareas algorítmicas secuenciales específicas, debido a su hardware fijo y optimizado.[46, 47] Los DSP de punto fijo, en particular, consumen menos energía y son más económicos que sus homólogos de punto flotante.[20]
- **FPGA:** Históricamente, las FPGA han sido vistas como dispositivos de alto consumo debido a la sobrecarga de su tejido configurable.[47, 48] Sin embargo, esta visión es a menudo obsoleta. Para tareas altamente paralelas, la capacidad de una FPGA para realizar más trabajo por ciclo de reloj puede resultar en una eficiencia de *rendimiento por vatio* superior.[48, 49, 50] Un estudio académico demostró que una implementación en FPGA a 100 MHz era 12.3 veces más rápida y consumía 2.4 veces menos energía que una implementación en un DSP a 1.5 GHz para la misma tarea de procesamiento de señales.[50] La gestión de la energía en una FPGA se puede optimizar mediante técnicas como el *clock gating* (desactivar el reloj en bloques no utilizados) y el uso

exclusivo de los recursos necesarios.[51]

La carga de trabajo de la IPU es mixta. Para las partes computacionalmente intensivas pero secuenciales de los algoritmos (por ejemplo, cálculos complejos dentro del BMS), un núcleo de procesador es más eficiente energéticamente. Para la lógica de control y el procesamiento de señales altamente paralelos (por ejemplo, la gestión simultánea de todas las E/S de sensores y la generación de PWM), la arquitectura FPGA puede ser más eficiente. Un DSP tendría que funcionar a una frecuencia de reloj muy alta para dar servicio secuencial a todas las tareas dentro del presupuesto de tiempo requerido, lo que aumenta su consumo de energía dinámico (donde la potencia es proporcional a la frecuencia, $P \propto f$).[51, 52] Una FPGA puede lograr el mismo rendimiento del sistema a una frecuencia de reloj mucho más baja al ejecutar las tareas en paralelo. Esto apunta de nuevo a un SoC como la solución óptima, donde los núcleos ARM pueden entrar en modos de bajo consumo cuando están inactivos, mientras que la lógica de control crítica en la PL funciona de manera continua y eficiente.

3.3. Idoneidad para Algoritmos de Control Avanzados (p. ej., Control Predictivo por Modelo, IA/ML)

La documentación de la IPU alude a la necesidad de "predicción de estados de conmutación óptimos" y algoritmos avanzados para el BMS, lo que abre la puerta a futuras implementaciones de control basado en Inteligencia Artificial (IA) y Aprendizaje Automático (ML).[1, 11]

- **DSP:** Son adecuados para algoritmos de control tradicionales como PID y pueden implementar técnicas avanzadas como el Control Predictivo por Modelo (MPC) y el Control por Modo Deslizante (SMC) en software.[53] Los DSP más recientes, como algunos de la serie C2000 de TI, están comenzando a integrar aceleradores de IA/ML (NPU) para mejorar estas capacidades.[21, 22]
- **FPGA:** Son una plataforma excelente para implementar algoritmos altamente paralelos como MPC o redes neuronales directamente en hardware.[51] Su arquitectura es una opción natural para las operaciones de matrices y vectores que son comunes en la inferencia de IA/ML, permitiendo un procesamiento en tiempo real con una latencia mínima.[54, 55, 56]
- **SoC (Zynq):** Ofrece la plataforma definitiva para la IA/ML en sistemas de control. El tejido de la FPGA (PL) puede usarse para crear aceleradores de hardware personalizados para la inferencia de redes neuronales, mientras que el procesador ARM (PS) ejecuta el código de la aplicación de nivel superior, un sistema operativo (por ejemplo, Linux) y los marcos de ML estándar.[31, 35, 36]

Para la IPU, la capacidad de un SoC para el co-diseño hardware/software es un diferenciador clave para el futuro. Un modelo de IA para predecir el SOH de la batería o para optimizar el rebote del "Pistón Atómico" podría acelerarse en el tejido de la FPGA, proporcionando resultados de inferencia en tiempo real a la aplicación de control principal que se ejecuta en el procesador ARM. Esta capacidad de "preparación para el futuro" asegura que la plataforma de la IPU no quede obsoleta, sino que pueda adoptar las innovaciones algorítmicas del mañana.

Sección 4: Viabilidad Económica y Ecosistema de Desarrollo

Esta sección pasa del rendimiento técnico a las consideraciones prácticas de coste y esfuerzo de desarrollo, que son críticas para un proyecto comercial como la IPU.

4.1. Análisis de Costes a Nivel de Componente y Sistema

Una comparación directa de los costes unitarios de los chips controladores puede ser engañosa; es crucial considerar el coste total del sistema y el coste total de propiedad (TCO).

- **DSP (TI C2000):** Los DSP de alto rendimiento como el TMS320F28379D tienen un precio unitario que oscila entre \$11 y \$20 USD en volúmenes de 1000 unidades (1kU).[57] Las placas de desarrollo, como la LAUNCHXL-F28379D, son relativamente asequibles, con un coste de entre \$40 y \$55 USD.[58, 59] Los modelos de gama más baja, como el F280049C, son más económicos, con precios de entre \$4 y \$9 USD (1kU), y sus placas de desarrollo LaunchPad tienen un coste similar.[60, 61, 62]
- **FPGA (AMD Spartan-7, Intel Cyclone 10 LP):** Las FPGA de bajo coste están disponibles por debajo de \$20-\$40 USD en unidades individuales para modelos como el XC7S15 o XC7S25.[63, 64] Las placas de desarrollo varían en precio, desde unos \$100 USD (Intel C10EFP-LP) hasta más de \$200 USD (Arty S7-50), con kits de evaluación de gama alta que pueden superar los \$800 USD (SP701).[63, 64, 65, 66, 67]
- **SoC (Zynq-7000):** La plataforma Zynq, al combinar un procesador y una FPGA, puede reducir el coste total de la lista de materiales (BOM) al integrar funciones que de otro modo requerirían múltiples chips externos, como lógica de interconexión o controladores de interfaz personalizados.[34, 40]

Aunque un DSP de gama media puede parecer la opción más económica en cuanto a coste unitario, el coste real es el del sistema. Una FPGA o un SoC pueden integrar lógica que de otro modo requeriría componentes externos, reduciendo potencialmente el BOM general. Además, el mayor rendimiento de una FPGA podría permitir el uso de componentes magnéticos más pequeños y baratos en la etapa de potencia debido a las mayores frecuencias de conmutación posibles, lo que reduce aún más el coste del sistema.[68, 69, 70]

Tabla 1: Análisis Comparativo de Costes de las Plataformas de Control (Precios de Unidades y Kits de Desarrollo)

La siguiente tabla resume los costes de hardware iniciales para la creación de prototipos y la producción en volumen, proporcionando una base para la evaluación económica.

Tipo de Plataforma	Dispositivo/Familia Específica	Kit de Desarrollo	Precio del Kit de Desarrollo (USD)	Precio Unitario (1kU, USD)	Características Clave	Fuentes
DSP	TI C2000 TMS320F28379D	LAUNCHXL-F28379D	~\$40 - \$55	~\$11 - \$12	Dual-core 200MHz C28x + CLA, FPU,	[57, 58, 59]

					HRPWM, ADCs rápidos	
DSP	TI C2000 F280049C	LAUNCHXL- F280049C	~\$40 - \$47	~\$4 - \$9	100MHz C28x + CLA, FPU, HRPWM, ADCs	[60, 61, 62]
FPGA	AMD Spartan-7 XC7S50	Arty S7-50	~\$203	~\$70	52k celdas lógicas, 120 slices de DSP	[63, 64]
FPGA	Intel Cyclone 10 LP	C10EFP-LP	~\$100	~\$20 - \$40 (estimado)	25k elementos lógicos	[65, 66]
SoC	AMD Zynq-7000 (Z-7010/20)	Digilent ZedBoard	~\$500+	~\$50 \$100+	- Dual-core ARM Cortex-A9 + FPGA (28k-85k celdas lógicas)	[32, 37]

Nota: Los precios son aproximados y están sujetos a cambios según el distribuidor y la configuración específica.

4.2. Flujo de Trabajo de Desarrollo, Cadenas de Herramientas y Tiempo de Comercialización

El flujo de trabajo y las herramientas de desarrollo son factores determinantes en el coste y el tiempo de un proyecto.

- **DSP:** El desarrollo se centra en software, utilizando lenguajes como C/C++ en Entornos de Desarrollo Integrados (IDE) como Code Composer Studio de TI.[71] Los fabricantes como TI proporcionan extensas bibliotecas de software (por ejemplo, la Digital Control Library), Kits de Desarrollo de Software (SDK) y diseños de referencia que pueden acelerar significativamente el desarrollo.[23, 71, 72] Este flujo de trabajo es familiar para la mayoría de los ingenieros de software embebido.
- **FPGA:** El desarrollo requiere lenguajes de descripción de hardware (HDL) como Verilog o VHDL y cadenas de herramientas complejas como AMD Vivado o Intel Quartus.[46] Esto representa un conjunto de habilidades diferente al de la programación de software y puede tener una curva de aprendizaje más pronunciada.[46, 73] Aunque existen herramientas de Síntesis de Alto Nivel (HLS) que convierten C/C++ a HDL, todavía están en proceso de maduración.[46, 48] El esfuerzo de desarrollo de una FPGA puede ser

hasta 5 veces mayor que el de un DSP para la misma aplicación.[48]

- **SoC (Zynq):** Requiere un flujo de trabajo híbrido. El software para el procesador ARM se desarrolla con herramientas C/C++ estándar, mientras que el hardware en la PL se desarrolla con HDL/HLS. Herramientas como MathWorks Simulink pueden automatizar este proceso generando tanto el código C como el HDL a partir de un único modelo de sistema, lo que agiliza el flujo y reduce los errores de codificación manual.[37, 42]

Para un equipo con una sólida experiencia en software embebido, el DSP ofrece la ruta de desarrollo más rápida y familiar. Sin embargo, para un sistema tan complejo como la IPU, un enfoque puramente de software podría no ser suficiente. La ruta de la FPGA requiere una experiencia especializada en HDL, lo que podría aumentar el tiempo y el coste del desarrollo. El SoC, especialmente cuando se combina con herramientas de diseño basadas en modelos como Simulink, ofrece un compromiso poderoso que puede acelerar el desarrollo de sistemas de control heterogéneos y complejos.

4.3. Disponibilidad de Talento y Requisitos de Experiencia

La elección de la plataforma también depende del equipo de desarrollo. Los ingenieros con dominio de C/C++ para microcontroladores y DSP son considerablemente más comunes que aquellos con una profunda experiencia en HDL y diseño de FPGA.[73] Sin embargo, algunos argumentan que el verdadero desafío no es aprender HDL, sino encontrar ingenieros de sistemas de control competentes, quienes podrían adquirir las habilidades de HDL con relativa rapidez.[74]

Esta es una consideración crítica de gestión de proyectos. Optar por un DSP minimiza la necesidad de nueva contratación o formación extensa, pero podría comprometer el rendimiento final de la IPU. Elegir una FPGA o un SoC puede requerir una inversión en talento o formación, pero probablemente conducirá a un producto más robusto y de mayor rendimiento. La disponibilidad de herramientas HLS y de diseño basado en modelos para SoC puede ayudar a cerrar esta brecha de experiencia.

El "coste" de una plataforma es, por tanto, una variable multidimensional que abarca no solo el precio del silicio, sino también el coste del desarrollo (herramientas y talento) y el coste de oportunidad asociado a un mayor tiempo de comercialización. Existe una compensación directa entre el coste inicial de los componentes y la complejidad y el riesgo del desarrollo. Para un sistema complejo y de alto rendimiento como la IPU, el coste unitario ligeramente superior de un SoC se justifica por la reducción del riesgo de desarrollo y el camino más rápido hacia una solución robusta y de alto rendimiento.

Sección 5: Modularidad, Escalabilidad y Preparación para el Futuro

Esta sección evalúa las plataformas en función de su capacidad para adaptarse a los cambios, escalar para futuras versiones del producto e incorporar nuevas tecnologías, lo cual es esencial para el éxito a largo plazo de la línea de productos IPU.

5.1. Flexibilidad del Hardware y Reconfigurabilidad en Campo

La capacidad de adaptar y actualizar un producto después de su despliegue es una ventaja competitiva significativa.

- **DSP:** La flexibilidad se limita exclusivamente al software. Los algoritmos pueden actualizarse, pero los periféricos de hardware (número de PWM, canales de ADC, etc.) son fijos y se definen en el momento de la selección del chip.[18] Cualquier cambio que requiera una nueva interfaz de hardware implicaría un rediseño completo de la placa.
- **FPGA/SoC:** Ofrecen una flexibilidad sin parangón. La lógica programable puede ser reconfigurada en campo para actualizar la funcionalidad del hardware, no solo el software.[26, 75, 25] Esto permite añadir nuevas interfaces, implementar nuevos periféricos de control o cambiar por completo un algoritmo de hardware después de que el producto haya sido desplegado, a menudo a través de una actualización de firmware remota (FOTA).

Para la IPU, la reconfigurabilidad de una FPGA/SoC es una ventaja estratégica. Si se necesita integrar un nuevo tipo de sensor o si se descubre que un esquema de PWM diferente es más eficiente, el hardware puede actualizarse remotamente. Un diseño basado en DSP estaría bloqueado en su conjunto de periféricos inicial, lo que podría requerir un costoso rediseño de hardware para una actualización de este tipo.

5.2. Escalabilidad del Rendimiento para Futuras Generaciones de IPU

Una plataforma de control debe permitir una ruta de escalado clara para una familia de productos con diferentes puntos de precio y rendimiento.

- **DSP:** La escalabilidad se logra típicamente migrando a un DSP más potente dentro de la misma familia de productos (por ejemplo, de un C2000 de un solo núcleo a uno de doble núcleo) o a uno con una velocidad de reloj más alta.[76, 77] Esto mantiene la compatibilidad del software, pero está limitado por la hoja de ruta de productos del proveedor.
- **FPGA/SoC:** La escalabilidad es mucho más granular. Un diseño puede migrarse a una FPGA/SoC más grande de la misma familia para obtener más celdas lógicas, slices de DSP y memoria BRAM, lo que permite algoritmos más complejos o más canales de control en paralelo.[32, 33] Un solo diseño de chip puede soportar el control de múltiples ejes simplemente instanciando el núcleo IP de control varias veces.[41]

La naturaleza modular del diseño de FPGA/SoC es altamente propicia para crear una familia de productos IPU escalable. Un modelo base de IPU podría usar un SoC más pequeño y optimizado en costes. Un modelo de alto rendimiento podría usar un SoC más grande de la misma familia, reutilizando exactamente los mismos bloques de IP de hardware y software, pero añadiendo más características o canales de control en el espacio lógico adicional. Esto ofrece un nivel de reutilización de diseño y escalabilidad que es difícil de lograr con los DSP de arquitectura fija.

5.3. Adaptabilidad a Algoritmos y Componentes de Sistema en Evolución

La IPU operará en un ecosistema tecnológico que evoluciona rápidamente. El controlador debe ser capaz de adaptarse a estos cambios.

- **Nuevos Semiconductores de Potencia:** La IPU utilizará semiconductores de potencia avanzados como el Carburo de Silicio (SiC) y el Nitruro de Galio (GaN). Estos dispositivos permiten frecuencias de conmutación muy altas y tienen requisitos de

control de puerta exigentes y de muy baja latencia, una tarea para la que la lógica de una FPGA está perfectamente adaptada.[78, 79, 80, 81, 69, 70, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92]

- **Algoritmos Avanzados:** Los algoritmos de control también están evolucionando, con una clara tendencia hacia técnicas basadas en IA/ML para el control predictivo y el mantenimiento predictivo.[54, 55, 7, 10, 56, 93, 94]

La arquitectura SoC proporciona la plataforma ideal para esta evolución. Los algoritmos pueden ser prototipados y ejecutados en el procesador ARM, con secciones críticas para el rendimiento aceleradas en la lógica programable. Esto asegura que la plataforma IPU no esté anclada a los algoritmos de hoy, sino que esté "preparada para el futuro" para adoptar las innovaciones del mañana, garantizando una vida útil más larga del producto y una ventaja competitiva sostenida. La modularidad de una arquitectura FPGA/SoC no es solo una característica técnica, sino una ventaja empresarial estratégica. Permite un enfoque de diseño "basado en plataforma" para la línea de productos IPU, lo que reduce los costes de I+D a largo plazo y acelera el lanzamiento de nuevas variantes de productos. Con un enfoque basado en SoC, la lógica de control (por ejemplo, para la generación de PWM, interfaces de sensores) puede desarrollarse como núcleos de IP reutilizables y parametrizables en el tejido de la FPGA.[41, 95] Para crear una nueva variante de IPU, un diseñador puede simplemente instanciar más o menos de estos núcleos de IP en un SoC de diferente tamaño dentro de la misma familia (por ejemplo, un Zynq-7000S para una versión de bajo coste, un Zynq-7020 para una versión de alto rendimiento).[32, 33] El software de alto nivel que se ejecuta en el procesador ARM requeriría cambios mínimos, ya que interactúa con estos núcleos de IP a través de interfaces estandarizadas como AXI.[42] Esto crea una plataforma altamente modular y escalable, estableciendo un vínculo causal directo: **modularidad de SoC → alta reutilización de IP → reducción del esfuerzo de desarrollo para nuevas variantes → menor tiempo de comercialización y menores costes de I+D a largo plazo.**

Sección 6: Síntesis y Recomendación Final para la IPU

Esta sección final sintetiza los hallazgos del análisis precedente para proporcionar una recomendación clara y basada en la evidencia para el controlador táctico de la IPU.

6.1. Resumen de las Compensaciones para la Aplicación IPU

La elección del controlador táctico para la IPU implica una cuidadosa evaluación de las compensaciones entre rendimiento, coste y modularidad. Un DSP ofrece un bajo coste unitario y un flujo de desarrollo familiar, pero su arquitectura secuencial presenta riesgos de rendimiento y cuellos de botella al gestionar las múltiples tareas de control paralelas y de tiempo crítico de la IPU. Una FPGA pura ofrece un paralelismo y un determinismo inigualables, resolviendo el problema del control concurrente a nivel de hardware, pero a costa de una mayor complejidad de desarrollo y un coste unitario potencialmente más alto.

La arquitectura SoC emerge como la solución que mejor equilibra estas compensaciones. Aprovecha la fortaleza de la FPGA para el control de tiempo real estricto y la flexibilidad de un procesador para la gestión de sistemas complejos. Aunque el coste unitario de un SoC puede ser superior al de un DSP, su capacidad para reducir el BOM del sistema, mitigar los riesgos

de rendimiento y acelerar el tiempo de comercialización para una familia de productos completa lo convierte en la opción más rentable desde una perspectiva de ciclo de vida del producto.

Tabla 2: DSP vs. FPGA vs. SoC: Resumen de las Compensaciones para la Aplicación IPU

La siguiente tabla consolida el análisis comparativo, evaluando cada arquitectura frente a los criterios más críticos para la IPU.

Criterio de Evaluación	DSP (p. ej., TI C2000)	FPGA	SoC (p. ej., AMD Zynq-7000)
Rendimiento en Tiempo Real (Latencia/Determinismo)	Alto: Optimizado para baja latencia en un solo bucle, pero el rendimiento puede degradarse con múltiples tareas concurrentes debido a la naturaleza secuencial y las interrupciones.	Excelente: Latencia ultrabaja y determinismo a nivel de ciclo de reloj. Ideal para bucles de control de tiempo real estricto como el "Pistón Atómico".	Excelente: Combina lo mejor de ambos: la PL de la FPGA para un control determinista de tiempo real estricto y el PS para tareas de tiempo real flexible.
Capacidad de Procesamiento Paralelo	Baja: Procesamiento secuencial. La concurrencia se simula a través de la conmutación de tareas, lo que puede crear cuellos de botella.	Excelente: Paralelismo de hardware verdadero. Múltiples bucles de control se ejecutan de forma independiente y simultánea sin contención de recursos.	Excelente: La PL ofrece paralelismo de hardware real para los bucles de control, mientras que el PS de doble núcleo puede ejecutar tareas de software en paralelo.
Eficiencia Energética (Rendimiento/Vatio)	Buena: Muy eficiente para algoritmos secuenciales específicos. El consumo aumenta con la frecuencia de reloj necesaria para gestionar múltiples tareas.	Buena a Excelente: Puede ser más eficiente que un DSP para cargas de trabajo altamente paralelas, ya que puede lograr un mayor rendimiento a una frecuencia de reloj más baja.	Excelente: Permite la optimización energética a nivel de sistema, utilizando la PL eficiente para tareas constantes y poniendo el PS en modos de bajo consumo cuando está inactivo.
Flujo de Trabajo y Herramientas de Desarrollo	Sencillo: Ecosistema maduro basado en C/C++ e IDEs estándar.	Complejo: Requiere experiencia en HDL (Verilog/VHDL).	Híbrido: Requiere tanto habilidades de software (C/C++) como

	(p. ej., Code Composer Studio). Amplias bibliotecas disponibles.	cadenas de herramientas complejas (p. ej., Vivado). Curva de aprendizaje pronunciada.	de hardware (HDL). El diseño basado en modelos (p. ej., Simulink) puede simplificar y unificar el flujo de trabajo.
Experiencia Requerida	Común: Amplia disponibilidad de ingenieros de software embebido con experiencia en C/C++.	Especializada: Menor disponibilidad de ingenieros con experiencia profunda en diseño de FPGA y HDL.	Multidisciplinar: Requiere una combinación de experiencia en software y hardware, o un equipo con ambas habilidades.
Coste Unitario	Bajo a Medio: Generalmente la opción más económica a nivel de componente.	Medio a Alto: El coste unitario es típicamente más alto que el de un DSP de rendimiento comparable.	Medio a Alto: El coste unitario es más alto que el de un DSP, pero competitivo al considerar la integración de funciones.
Coste a Nivel de Sistema (BOM)	Medio: Puede requerir componentes externos adicionales para lógica de interfaz o protección.	Bajo a Medio: Puede integrar lógica de pegamento y periféricos personalizados, reduciendo el número de componentes externos.	Bajo: La alta integración de procesador, FPGA y periféricos en un solo chip reduce significativamente el BOM total.
Modularidad y Escalabilidad	Media: La escalabilidad se limita a la hoja de ruta de productos del proveedor. La reutilización del software es alta dentro de la misma familia.	Excelente: Altamente modular. Los diseños de IP se pueden reutilizar y escalar fácilmente a FPGAs más grandes para aumentar el rendimiento o las características.	Excelente: Ofrece una plataforma escalable. Se pueden crear familias de productos utilizando el mismo software en el PS y escalando los recursos de la PL.
Preparación para el Futuro (Adaptabilidad IA/ML)	Media: Los nuevos DSP están incorporando aceleradores de IA, pero la flexibilidad es	Alta: La arquitectura paralela es ideal para acelerar algoritmos de IA/ML en hardware.	Excelente: La mejor plataforma para co-diseño HW/SW, permitiendo la aceleración de la

	limitada.		inferencia de IA en la PL mientras se ejecutan marcos de alto nivel en el PS.
--	-----------	--	---

6.2. La Arquitectura de Controlador Óptima para la IPU

Basado en el análisis exhaustivo de los requisitos de control únicos de la IPU y la evaluación comparativa de las arquitecturas candidatas, la recomendación inequívoca es la adopción de una **arquitectura de Sistema en un Chip (SoC), como la familia Zynq-7000 de AMD, como el controlador táctico para la Unidad de Potencia Inteligente.**

Esta recomendación se fundamenta en la capacidad superior del SoC para abordar el problema de control heterogéneo y multivelocidad de la IPU. Proporciona el rendimiento determinista, de baja latencia y de procesamiento paralelo de una FPGA para los bucles de control más críticos del sistema (el "Pistón Atómico" y la conmutación de potencia), al tiempo que ofrece la flexibilidad y el potente ecosistema de software de un procesador ARM para la gestión de sistemas de nivel superior, las comunicaciones y la evolución algorítmica futura. Este enfoque híbrido mitiga los riesgos de rendimiento y desarrollo de manera más efectiva que una solución pura de DSP o FPGA, ofreciendo el mejor equilibrio entre rendimiento, rentabilidad a largo plazo y modularidad.

6.3. Hoja de Ruta Estratégica de Implementación

Para una implementación exitosa y eficiente de la solución SoC recomendada, se propone la siguiente hoja de ruta estratégica:

1. **Fase 1: Prototipado y Reducción de Riesgos:** Utilizar un kit de desarrollo basado en Zynq-7000 (por ejemplo, el Avnet Intelligent Drives Kit o el Digilent ZedBoard) [32, 37, 42] en conjunto con un flujo de diseño basado en modelos (por ejemplo, MATLAB/Simulink). Este enfoque permite un prototipado rápido de los algoritmos de control y una validación temprana de la partición hardware/software, identificando y mitigando los riesgos técnicos en las primeras etapas del proyecto.[37, 42]
2. **Fase 2: Desarrollo de Núcleos IP:** Desarrollar las funciones de control críticas (generación de PWM, interfaces de sensores, máquina de estados del "Pistón Atómico") como núcleos de Propiedad Intelectual (IP) reutilizables y parametrizables en la lógica programable (PL). Esto crea bloques de construcción modulares para futuros diseños.
3. **Fase 3: Integración de Software y Sistema:** Desarrollar la aplicación de nivel superior (BMS, MPPT, comunicaciones, diagnósticos) en el procesador ARM (PS), ejecutando un sistema operativo adecuado (por ejemplo, Linux para una conectividad y flexibilidad ricas, o un RTOS como FreeRTOS para un determinismo más estricto). Integrar el software con los núcleos de IP de hardware a través de la interfaz de bus AXI de alto rendimiento.
4. **Fase 4: Escalado de la Familia de Productos:** Aprovechar la IP y el software desarrollados para escalar el diseño a diferentes puntos de coste y rendimiento. Al migrar a diferentes dispositivos dentro de la familia Zynq-7000 (por ejemplo, de un

Zynq-7000S de un solo núcleo a un Zynq-7020 de doble núcleo y mayor capacidad de PL), se puede lanzar una familia completa de productos IPU de manera rápida y rentable, maximizando la reutilización del diseño y minimizando el tiempo de comercialización.