

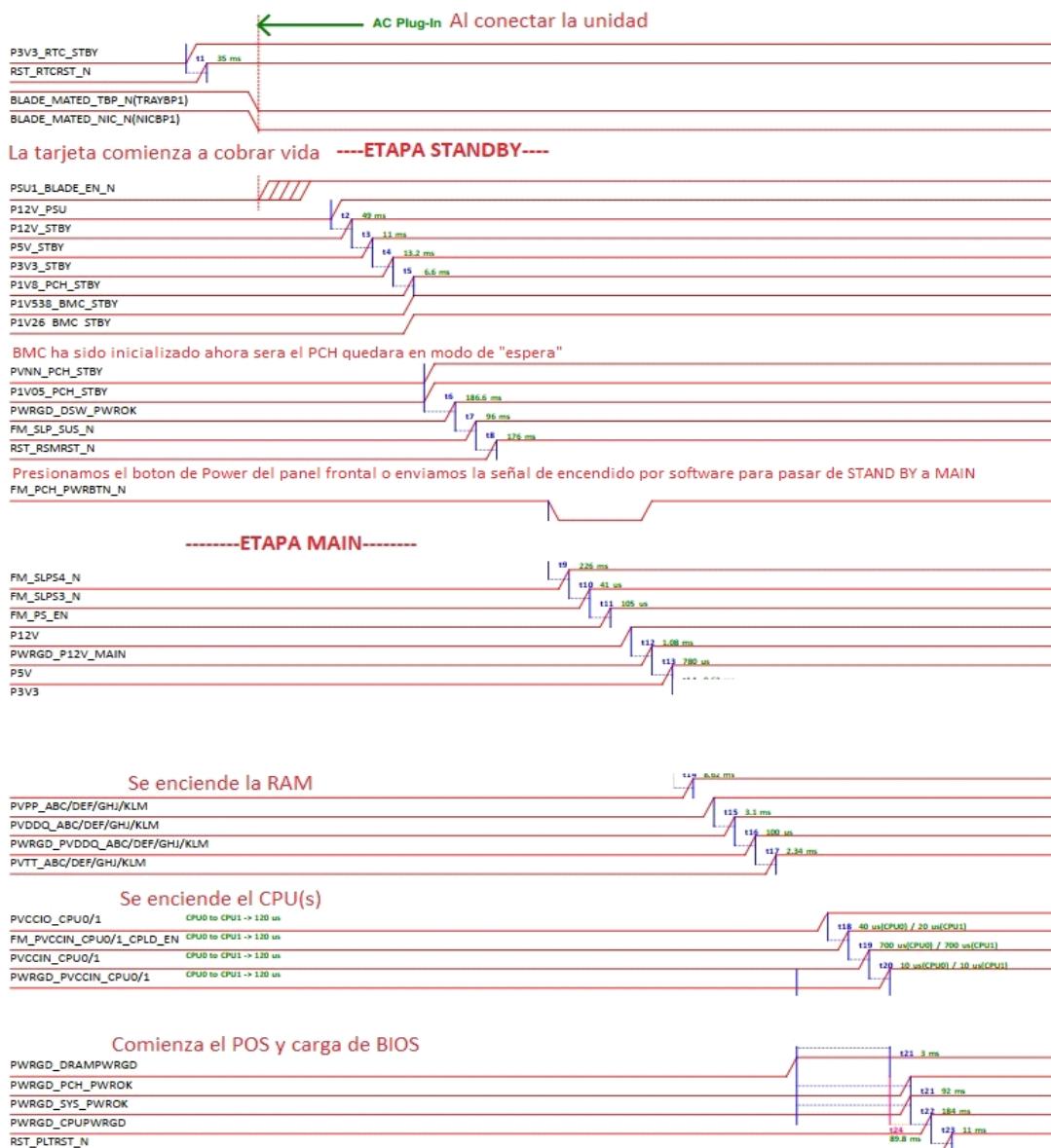
POWER ON SEQUENCE

COMENZAREMOS LA INTRODUCCION SIGUIENDO LA SECUENCIA DE "POWER ON" DEL ESQUEMATICO

Dividideros la secuencia en tres etapas, PLUG IN, STAND BY y MAIN, a su vez cada una estara

dividida en etapas correspondientes a la iniciacion de uno o varios IC que realicen una tarea del sistema y que esten estrechamente relacionados:

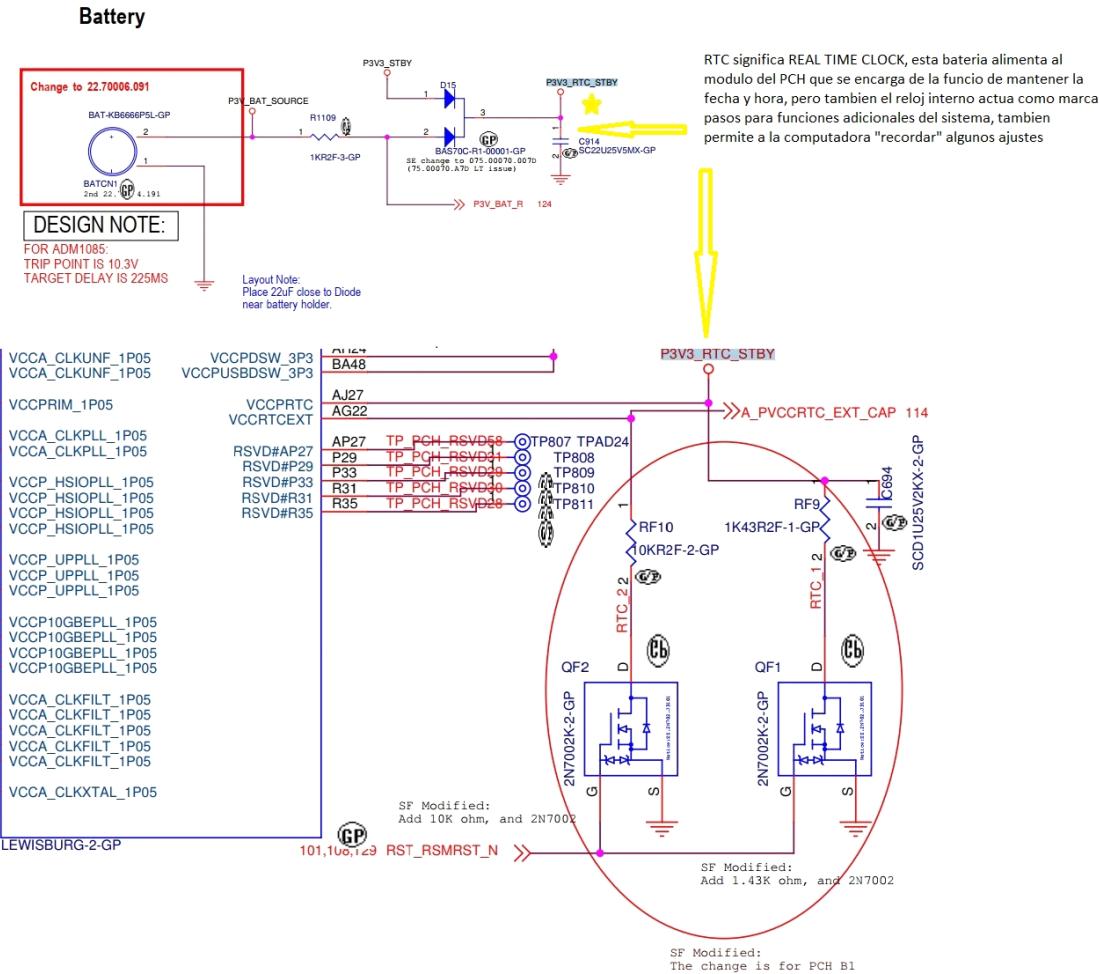
POWER ON SEQUENCE



AC PLUG IN

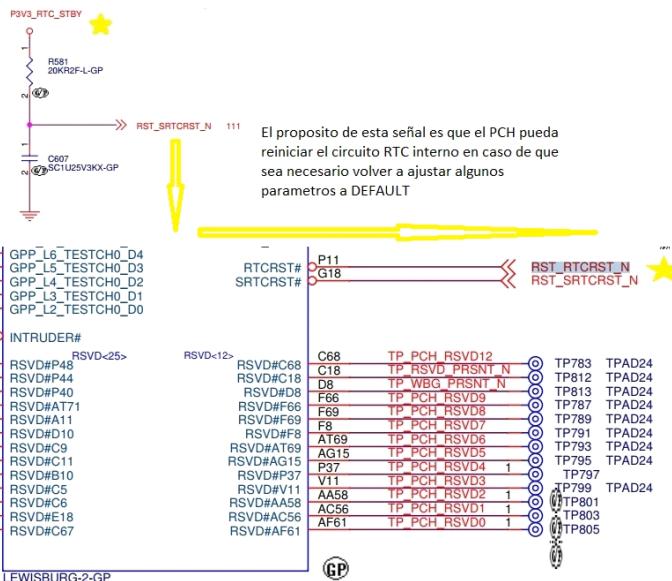
Estas primeras dos condiciones de POWER UP ocurren tras que la unidad sea conectada a la alimentacion de AC (recive la alimentacion de las fuentes externas), y son un punto importante de configuracion inicial.

P3V3_RTC_STBY

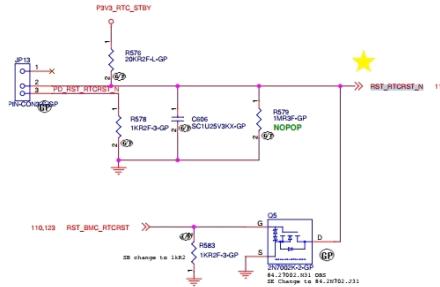


RST_RTCRST_N

A partir de la linea de alimentacion proporcionada por la bateria se crea mediante una divisor de voltage el valor logico de esta señal



CMOS CLEAR JUMPER

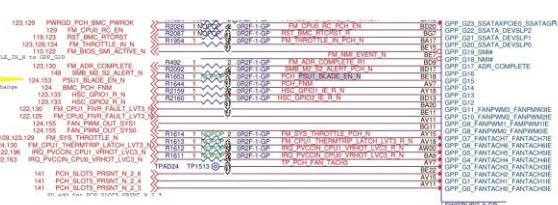
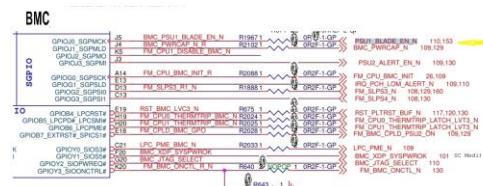


Tambien existe la opcion de realizar el proceso de borrado y ajuste a DEFAULT manualmente mediante un conector, ajustando un jumper o podemos ver que es posible que el BMC tambien realice esta funcion por si solo

STAND BY

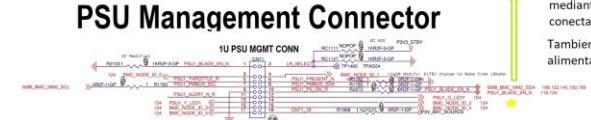
Cuando la unidad es conectada a la fuente de alimentacion AC, las lineas PSUx_BLADE_EN_N indican cuantas fuentes tiene el sistema

PSU1 BLADE EN N



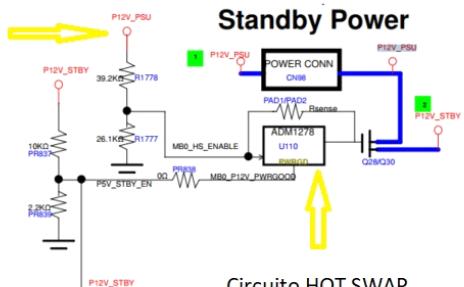
La fuente de alimentación externa, proporciona a la tarjeta madre alimentación mediante un conector, esta señal indica al PCH y al BMC que la unidad está conectada a la fuente de alimentación de AC.

Tambien sirve para detectar cuales de las fuentes conectadas a la unidad estan alimentadas a la fuente de AC

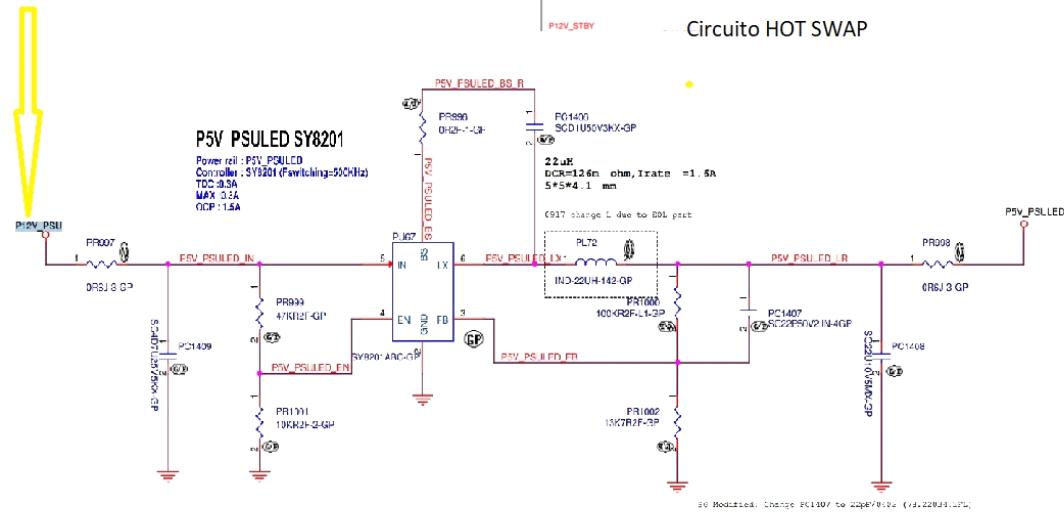


P12V_PSU

La linea de alimentacion P12V_PSU vista desde el punto de vista del sistema es como el cordon que va al toma corriente, solo pocos circuitos tomaran su energia de esta linea, entre ellos estan los LEDs indicadores de estado, los cuales deben proporcionar informacion del estado del sistema antes de que este "prenda", la funcion principal es alimentar al circuito HOT SWAP del servidor



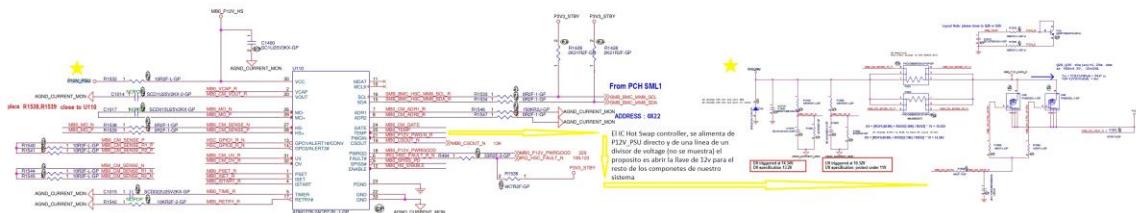
Circuito HOT SWAP



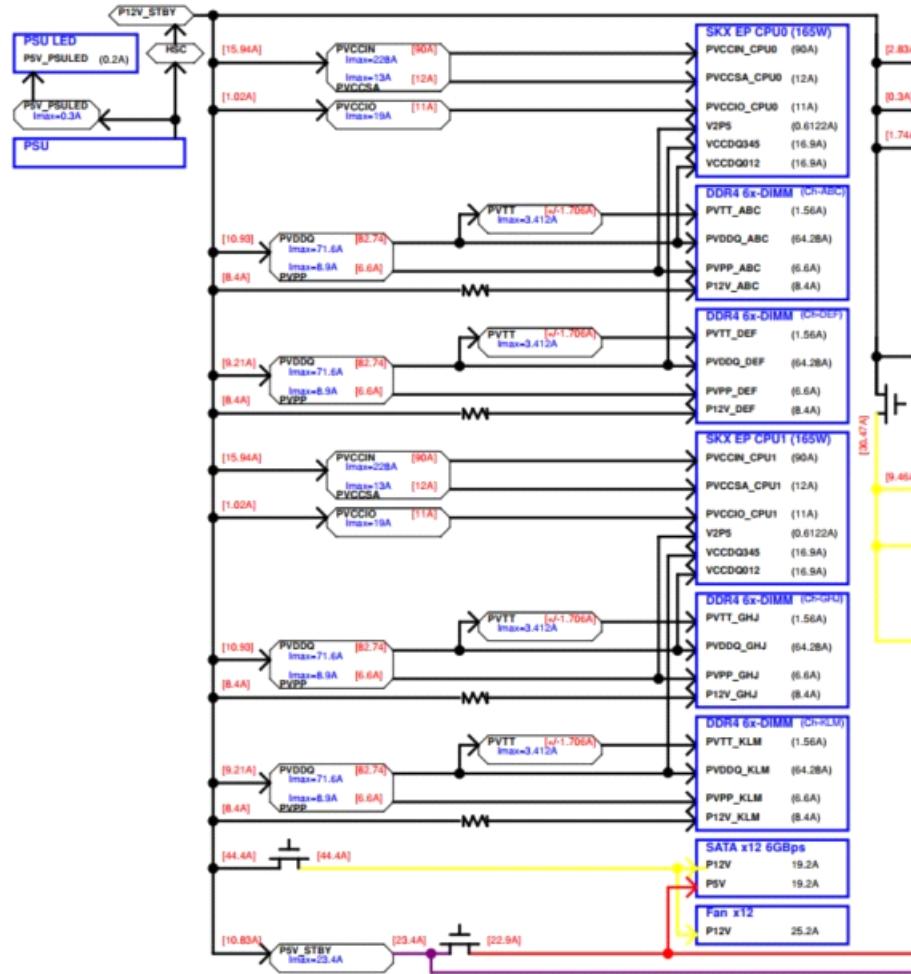
Hasta este punto todas estas acciones han ocurrido simplemente al conectar la unidad a la fuente de alimentacion AC (en caso de los BLADE SERVERS) y tras encender el switch (en servidores STAND ALONE).

P12V_STBY

La linea no es mas que continuacionde los P12V_PSU, pero tras pasar la "llave paso" de dos Fets de potencia controlados por el HOT SWAP, este IC tiene el proposito de detectar cualquier cambio en el nivel de voltage que pueda ser nocivo para el sistema y desconecta-conecta automaticamente los rieles de alimentacion.

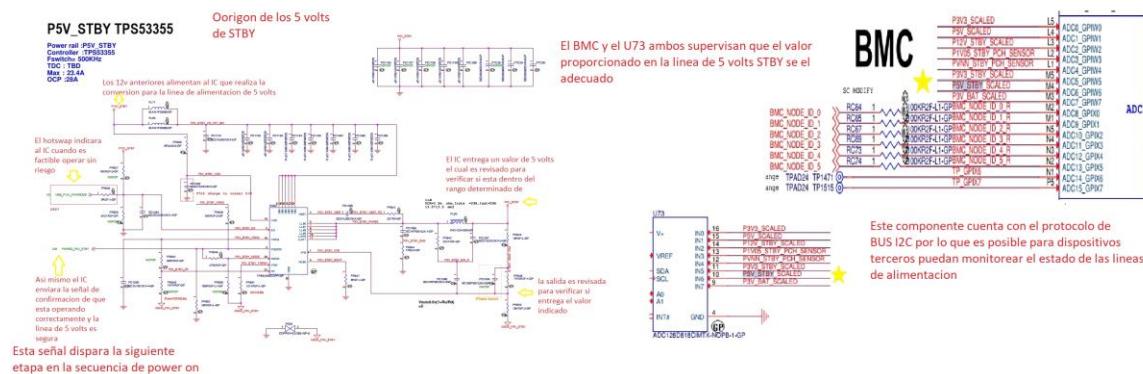


De esta manera pueden protegerse los demás dispositivos del servidor contra una falla en la linea de alimentacion principal:



P5V_STBY

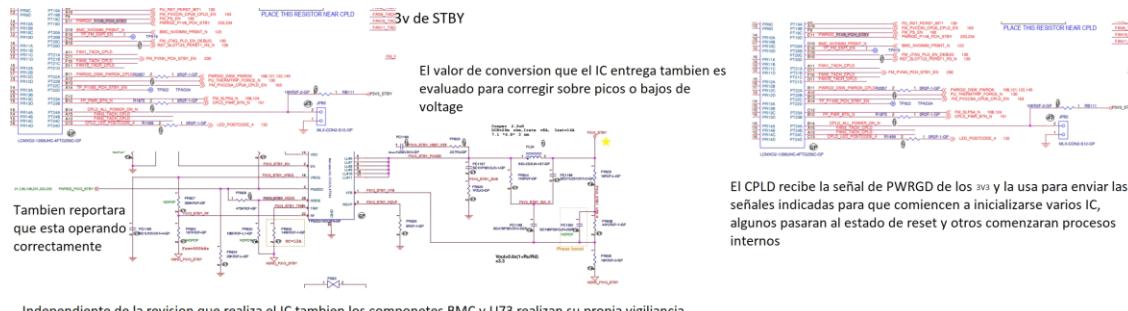
Cada uno de los IC que estan en la imagen anterior, realizaran una conversion de 12v a un valor requerido para que puedan operar los diferentes componentes, entrando a operar cuando sea su turno dentro de la secuencia de POWER ON y reportara un estatus de correcto para continuar con la siguiente etapa



para asegurar que la operacion de las lineas de alimentacion ocurren sin problema podemos el BMC y un segundo componente monitorearan que los valores esten dentro de los rangos de operacion.

P3V3_STBY

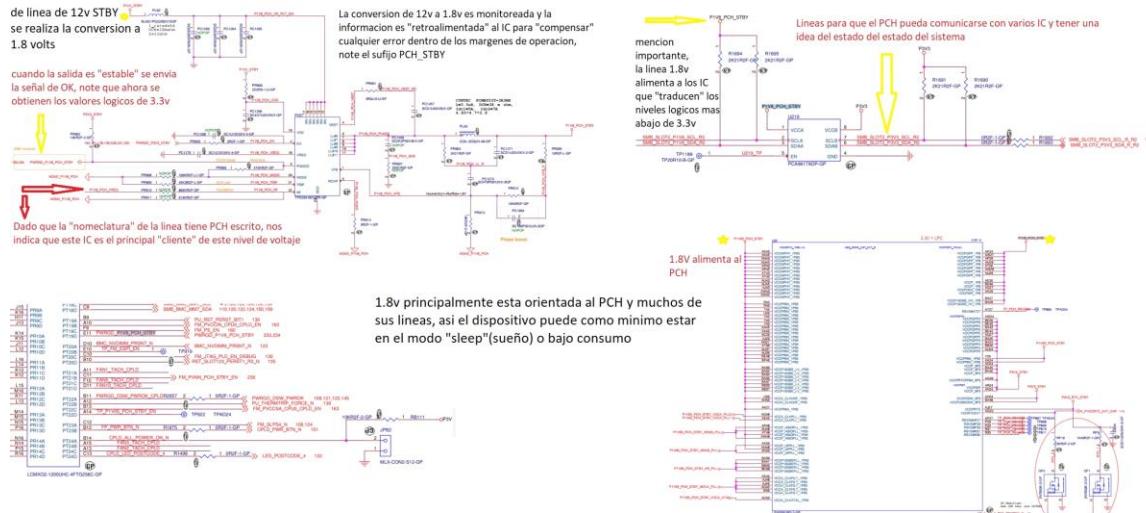
Siguiendo la secuencia de POWER ON y tomando en cuenta lo mencionado con anterioridad podemos ver la logica usada para inicializar al sistema.



La linea de alimentacion de 3.3 volts es muy importante porque alimenta muchos componentes y a su vez es un nivel logico muy usado, por esta razon la señal de PWRGD de 3.3 volts llega al CPLD el cual se encargara de disparar las señales correspondientes para comenzar a inicializar o sacar de reset varios IC, por el momento el lector debe quedarse con la idea de que la linea de alimentacion de 3.3V esta activa.

P1V8_PCH_STBY

Ahora que tenemos al CPLD alimentado y una linea logica de voltaje, podemos comenzar a encender a los componentes principales del sistema, entre ellos el PCH y comenzar a inicializar algunos servicios al menos a un nivel minimo.

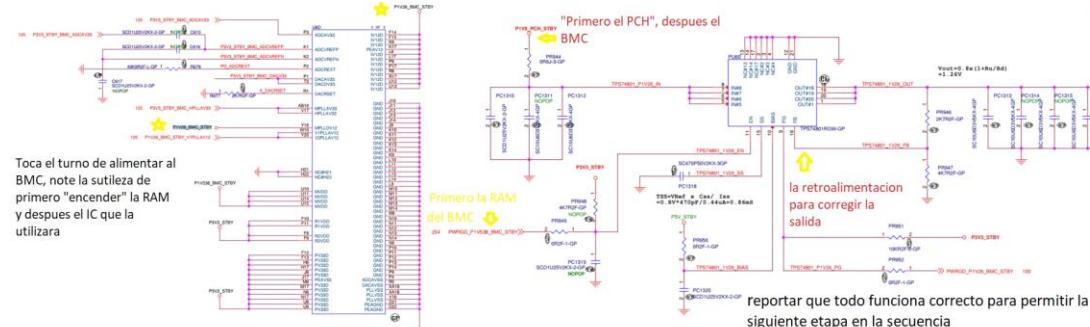


P1V538_BMC_STBY

Apesar de tener la palabra BMC en su nomenclatura, esta linea, no alimenta directamente al BMC si no a la memoria RAM externa del BMC.

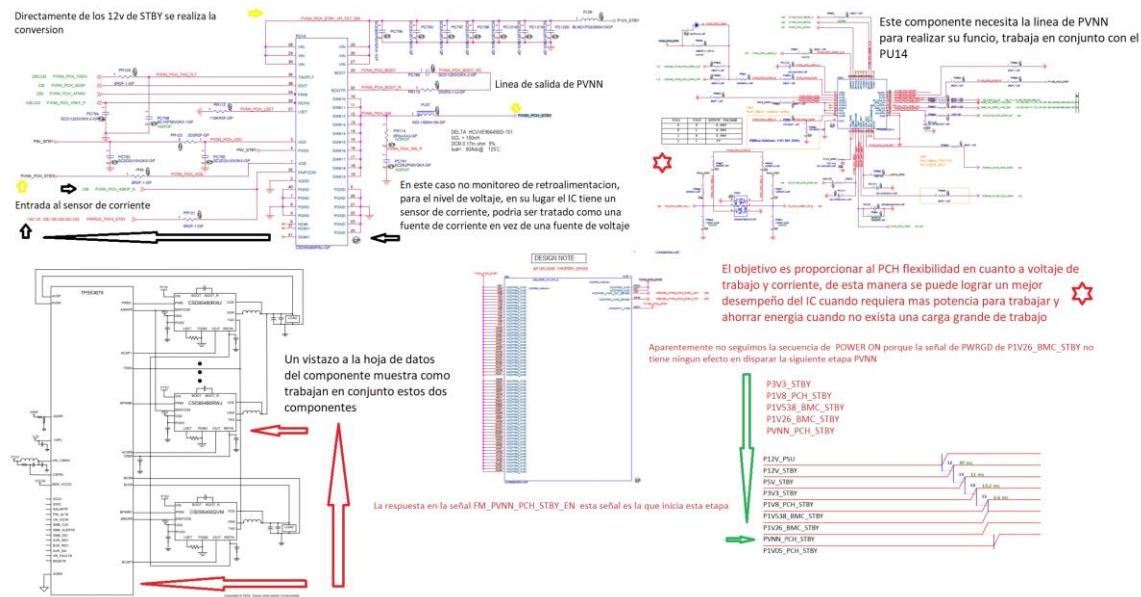


P1V26_BMC_STBY



Lo usual con esta linea de alimentacion, el lector debe de ir notando un patron.

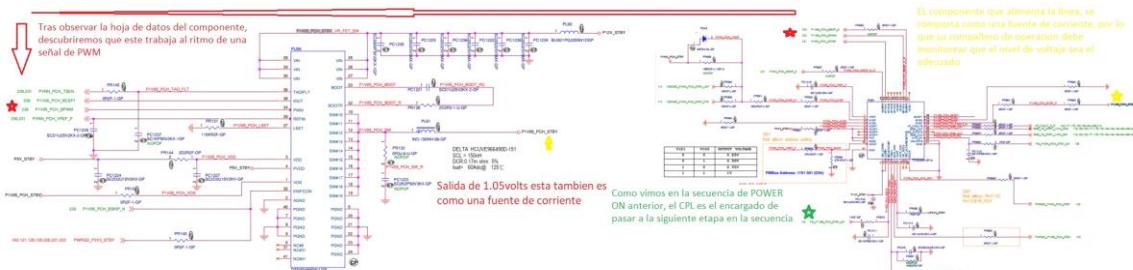
PVNN_PCH_STBY



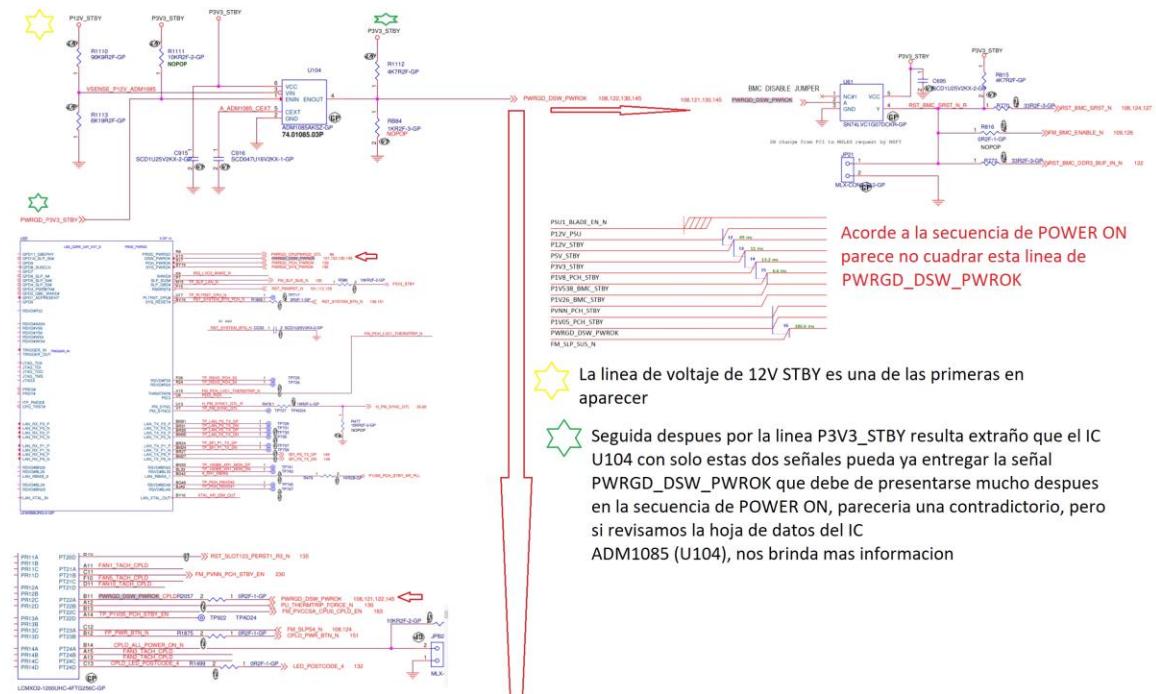
Hasta ahora existia un patron claro, la secuencia simplemente avanzaba en "cascada", pero la señal PWGD de P1V26_BMC_STBY, no activa ningun regulador de voltaje, y no parece que exista alguna señal que active al PU14 el cual genera la salida PVNN_PCH_STBY, la respuesta esta en la señal FM_PVNN_PCH_STBY_EN, el lector debe tomar nota de esta "peculiaridad" ya que se retomara mas adelante.

P1V05_PCH_STBY

Nuevamente el CPLD es el encargado de disparar esta linea de alimentacion para continuar con la secuencia, para informacion adicional sobre como los dos componentes el PU56 y el PU54 interactuan, revise la hoja de datos.



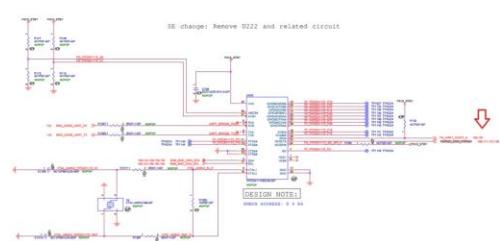
PWRGD_DSW_PWROK



Acorde a la secuencia de POWER ON parece no cuadrar esta linea de PWRGD_DSW_PWROK

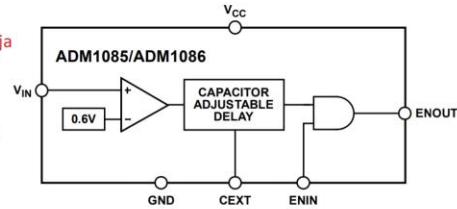
La linea de voltaje de 12V STBY es una de las primeras en aparecer

Seguida despues por la linea P3V3_STBY resulta extraño que el IC U104 con solo estas dos señales pueda ya entregar la señal PWRGD_DSW_PWROK que debe de presentarse mucho despues en la secuencia de POWER ON, pareceria una contradiccion, pero si revisamos la hoja de datos del IC ADM1085 (U104), nos brinda mas informacion



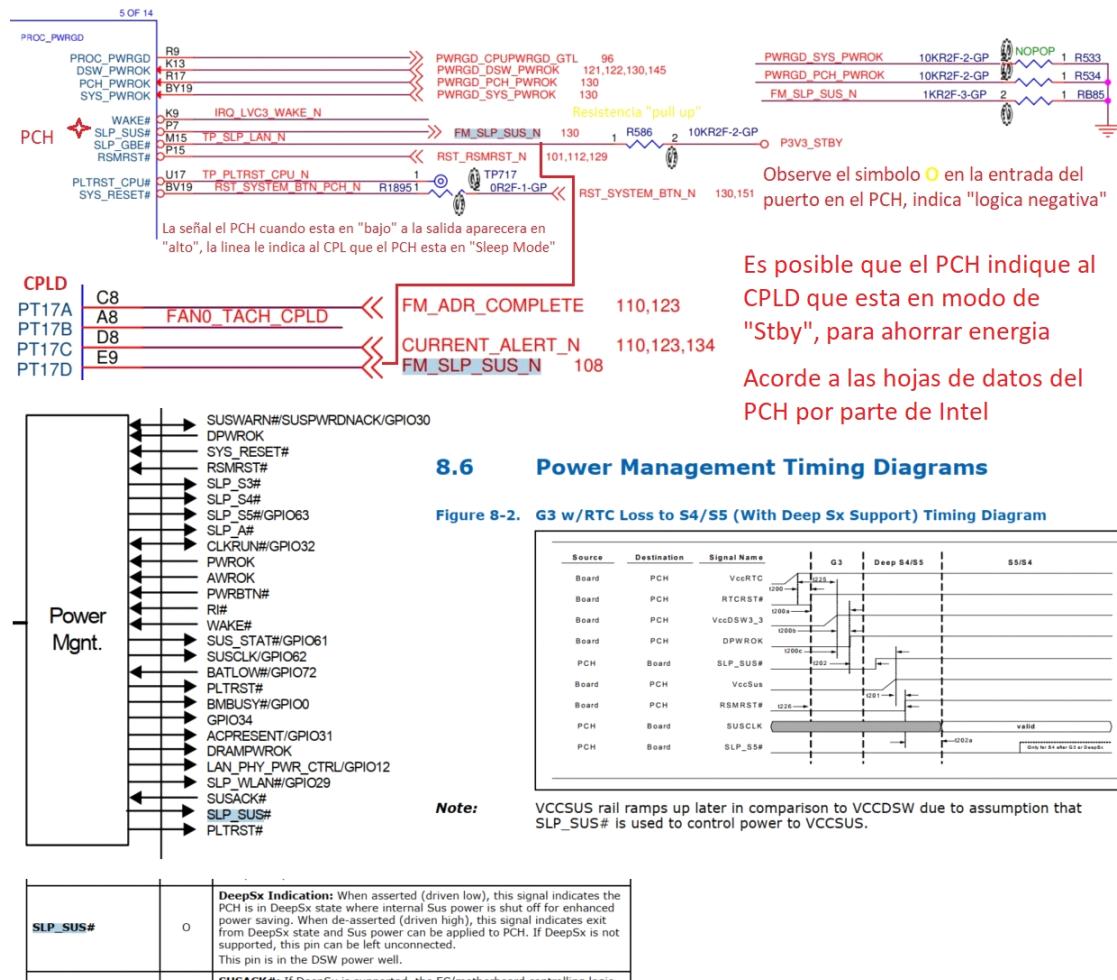
Un vistazo a la hoja de datos del IC que genera la señal nos da mas informacion cuando algo no parece claro

FUNCTIONAL BLOCK DIAGRAMS



Cuando algo no cuadre con la informacion presentada en el esquematico siempre es mejor revisar las hojas de datos como primera fuente de informacion adicional, podemos darnos cuenta que desde que el HOT SWAP inicializa la linea de 12V STBY, hasta la linea de PWRGD_DSW_PWROK, ya han pasado 325 ms aproximadamente (tiempo T2 hasta T6 en la secuencia), y para este punto el PCH deberia estar en la etapa de STBY (bajo consumo), el proposito del IC ADM1085, es retrasar la señal PWRGD_DSW_PWROK, hasta cuando "le toque entrar en escena".

FM_SLP_SUS_N

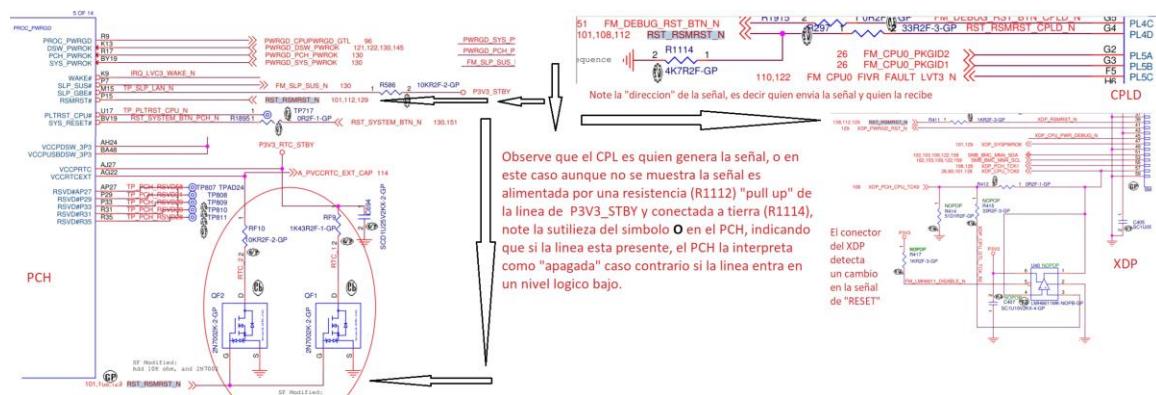


Para mayor informacion del propósito de esta linea revise la hoja de datos del PCH, por el momento simplemente nos quedamos con la idea de que la linea activa un "modo de operacion del PCH", probablemente indique al CPLD que este se encuentra en STBY, o en algun modo de ahorro de energia en espera de ser completamente activado.

Es posible que el PCH indique al CPLD que esta en modo de "Stby", para ahorrar energia

Acorde a las hojas de datos del PCH por parte de Intel

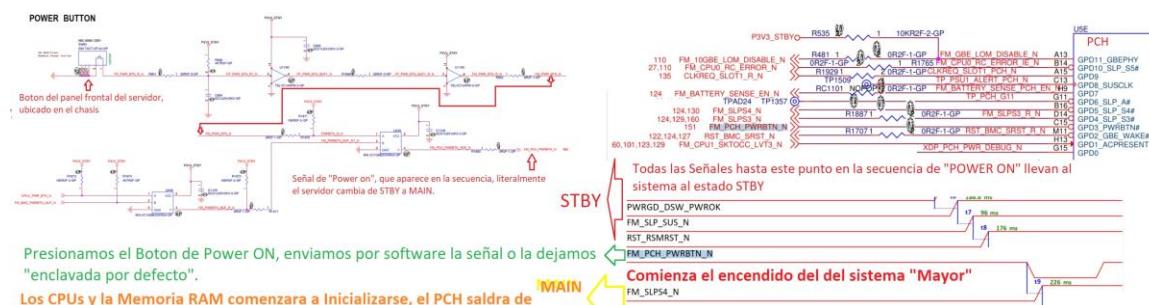
RST_RSMRST_N



Podemos interpretar esta señal como que el CPLD "saca" de "Reset" al PCH, y entonces comienza a "Inicializarse" (proceso de Booteo).

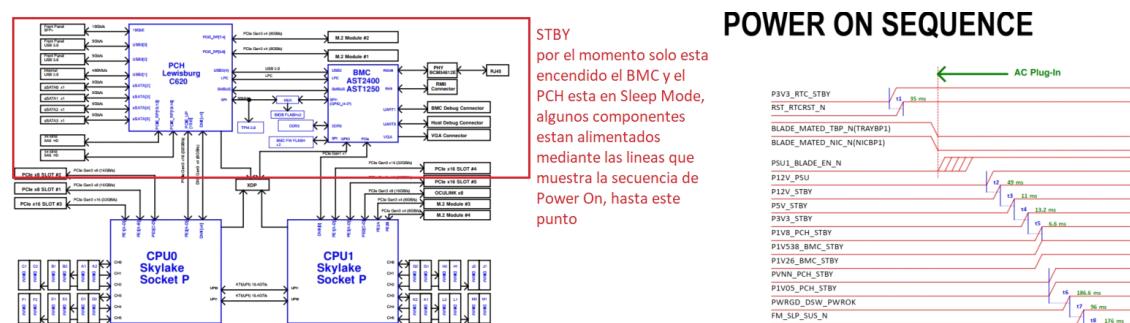
BOTON DE ENCENDIDO

FM_PCH_PWRBTN_N



Para este servidor al conectarlo a la alimentacion no lo encendera inmediatamente, solo llegara a la fase de STBY ,note que los componentes U245 y U246 son compuertas logicas, por lo que es necesario que el CPLD y el BMC esten encendidos para que al presionar el boton del panel frontal (SWB1) el servidor "prenda".

SERVICIOS QUE PODEMOS OBTENER EN LA ETAPA "STBY"



Acorde a la secuencia de Power On y al diagrama de bloques, por el momento el PCH esta en STBY y el BMC se encuentra activo, junto a algunos componentes de esquema de I2C.

I2C BLOCK DIAGRAM -BMC/PCH

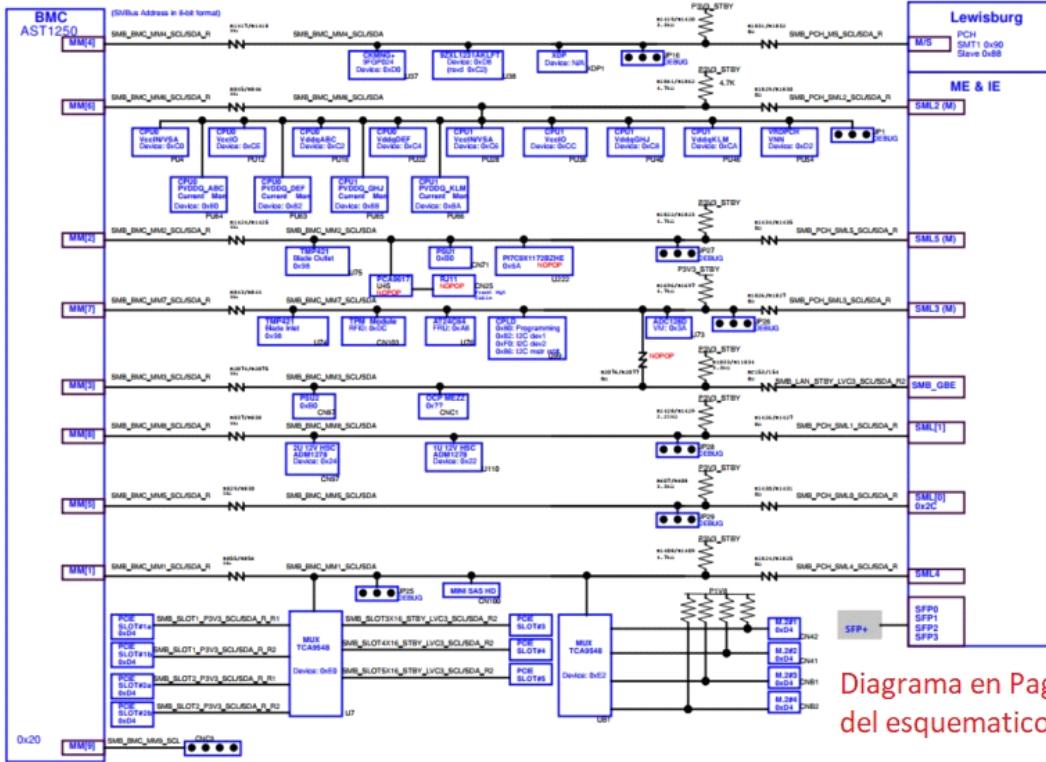
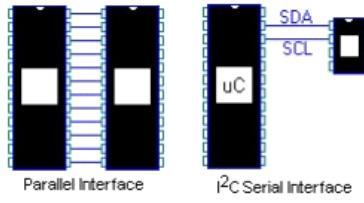


Diagrama en Pagina 6 del esquematico

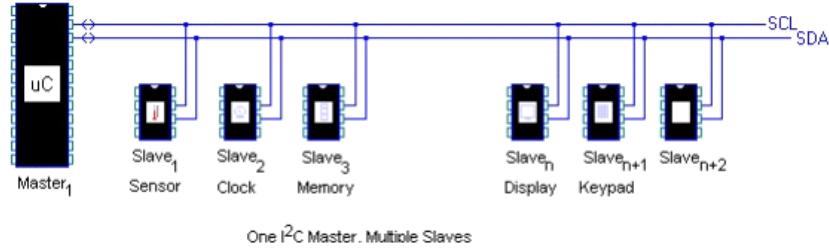
El I2C es un Protocolo de comunicacion diseñado por PHILIPS a principios de los años 80s, con el propósito de reducir los costos de manufactura de los productos electronicos, fue tanto su aceptacion que se volvio un estandar en la Industria.

La arquitectura del bus se muestra en la siguiente figura:

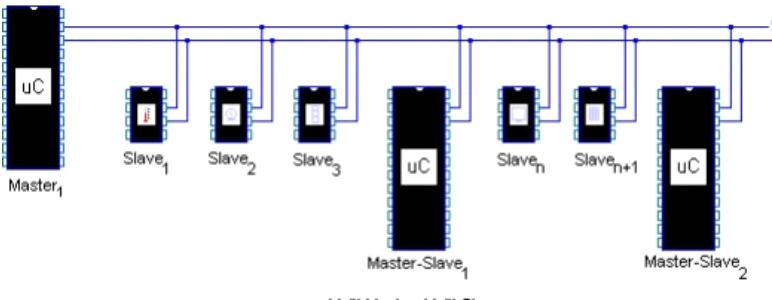


El objetivo del Protocolo es reducir el numero de lineas requerida para el intercambio de informacion entre componentes

La especificacion permite la conexion de varios componentes a un Bus



One I²C Master. Multiple Slaves

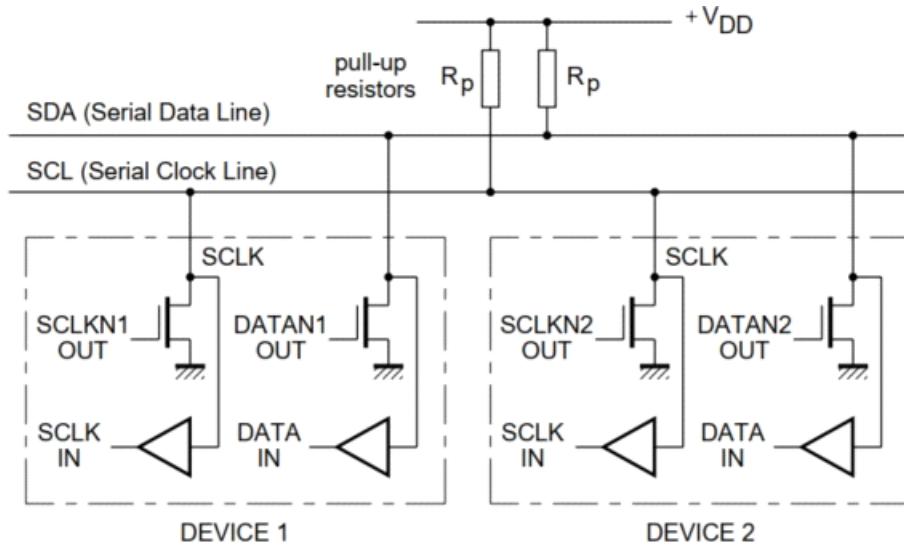


Tipicamente existe un IC Maestro y varios IC Esclavos

Para la configuracion de varios Maestros es necesario tomar algunas consideraciones y seguir la especificacion de Arbitraje de Bus

Hardware del Bus:

El hardware del bus se basa en una And cableada, las etapas de salida de los dispositivos conectados al bus deben ser drenador abierto, o colector abierto, para poder realizar la AND cableada:



Las salidas SDA y SCL estan conectadas a tension positivoa de alimentacion a traves de unas resistencias de Pull-Up, dependiendo del estado del transistor de salida de cada dispositivo puede ocurrir alguno de estos dos casos:

- Que el transistor este saturado, con lo cual lleva a nivel bajo o 0, a la linea correspondiente, independientemente del estado de los otros transistories. Es decir, el bust esta ocupado en nivel bajo.
- Que el transistor este en corte (estado de alta impedancia) con lo cual el estado de la linea depende de los otros transistores. Es decir, el bus esta libre y, si no hay ningun otro transistor saturado, la linea se encuentra en estado alto a traves de la resistencia Pull-Up conectada a la almimentacion.

El calculo de las resistencias de Pull-Up depende de la tension de alimentacion, de la capacidad del bus y del numero de dispositivos conectados.

Transferencia de Datos en el Bus

Para que la transferencia de información pueda iniciar el bus no debe estar ocupado. Esto quiere decir que los transistores de salida de todos los dispositivos conectados al bus I2C deben estar en alta impedancia. Para indicar que el bus está libre (no ocupado) las líneas de reloj (SCL) y datos (SDA) deben estar a nivel alto. Ocurrirán la siguiente secuencia de sucesos:

1.- Condición de start. SDA debe estar en flanco de bajada mientras que SCL permanece a nivel alto. Esta condición señala el comienzo de la transferencia de datos.

2.- Transmisión de los bits de información en cada pulso de reloj.

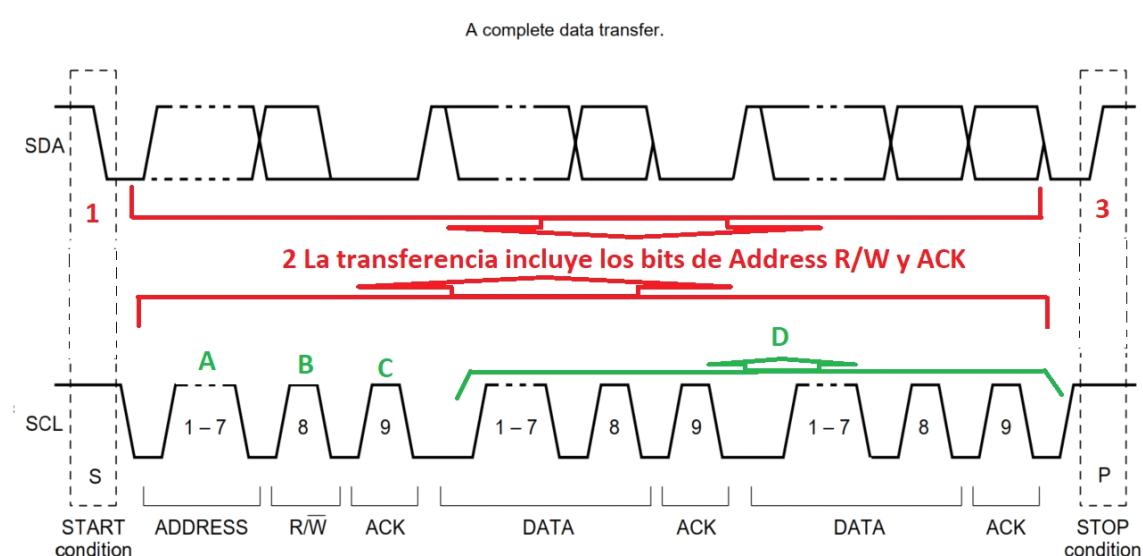
A-incluye los siete bits de direccionamiento de un dispositivo esclavo (ocurre una vez).

B-Un bit de lectura/escritura (R/W) que define si el esclavo es transmisor o receptor

C-un bit de reconocimiento para asegurar que el byte transferido fue procesado por el IC (ya sea Maestro o Esclavo).

D-Este proceso continúa hasta que la información haya sido transmitida, es decir el "Dialogo" entre el Maestro-Esclavo ha terminado.

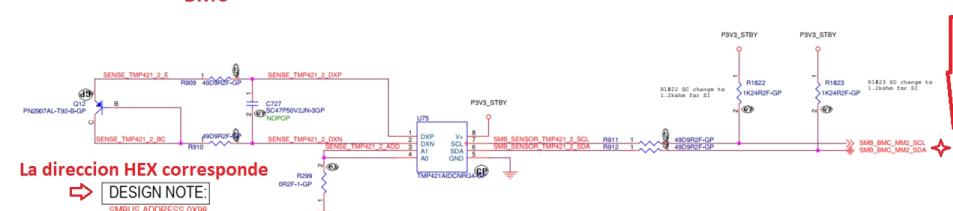
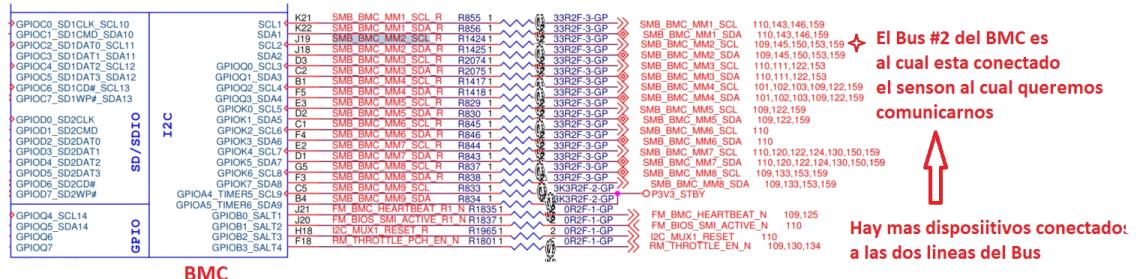
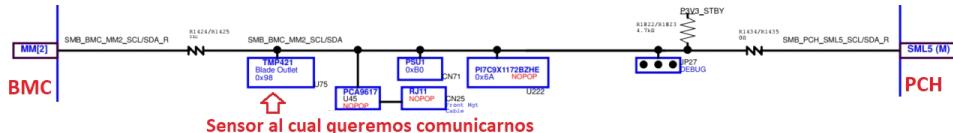
3.- Condición de Stop. en flanco de subida mientras que SCL permanece a nivel alto. Esta es la condición que indica el fin de la transferencia.



Tras terminar la Transferencia, es posible que otro dispositivo tome el control como nuevo maestro y elija a otro dispositivo como esclavo.

Algunos dispositivos esclavos típicos que pueden conectarse al Bus I2C, serían convertidores de A/D y D/A, expansores de pines de I/O de propósito general (GPIO), EEPROM, IC para monitoreo de sistemas (temperatura/voltaje), traductores de nivel de voltaje, para mayor información puede revisar el esquemático de I2C del servidor.

Tomemos de ejemplo al sensor de temperatura U75:



La direccion HEX corresponde

➡ DESIGN NOTE:
SMBUS ADDRESS 0x98

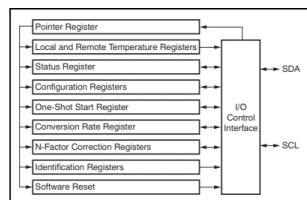


Figure 14. Internal Register Structure

A parte de la direccion del dispositivo tambien necesitamos conocer la estructura interna del mismo para poder extraer la informacion que necesitamos de sus registros internos, para esto necesitamos revisar la Hoja de Datos del IC,

El mapa de registros nos indica que 0x0100 hasta 0x0300 corresponde a registros de temperatura y estos contienen el byte de informacion del sensor

Table 3. Register Map

POINTER (HEX)	POR (HEX)	BIT DESCRIPTION								REGISTER DESCRIPTION
		7	6	5	4	3	2	1	0	
00	00	LT11	LT10	LT9	LT8	LT7	LT6	LT5	LT4	Local Temperature (High Byte) ⁽¹⁾
01	00	RT11	RT10	RT9	RT8	RT7	RT6	RT5	RT4	Remote Temperature 1 (High Byte) ⁽¹⁾
02	00	RT11	RT10	RT9	RT8	RT7	RT6	RT5	RT4	Remote Temperature 2 (High Byte) ^{(1) (2) (3)}
03	00	RT11	RT10	RT9	RT8	RT7	RT6	RT5	RT4	Remote Temperature 3 (High Byte) ^{(1) (2) (3)}
08	00	BUSY	0	0	0	0	0	0	0	Status Register
09	00	0	SD	0	0	0	RANGE	0	0	Configuration Register 1
0A	1C9C ⁽²⁾ /7C ⁽³⁾	0	REN3 ⁽³⁾	REN2 ^{(2) (3)}	REN	LEN	RC	0	0	Configuration Register 2
0B	07	0	0	0	0	0	R2	R1	R0	Conversion Rate Register
0F	X	X	X	X	X	X	X	X	X	One-Shot Start ⁽⁴⁾
10	00	LT3	LT2	LT1	LT0	0	0	PVLD	OPEN	Local Temperature (Low Byte)
11	00	RT3	RT2	RT1	RT0	0	0	PVLD	OPEN	Remote Temperature 1 (Low Byte)
12	00	RT3	RT2	RT1	RT0	0	0	PVLD	OPEN	Remote Temperature 2 (Low Byte) ^{(2) (3)}
13	00	RT3	RT2	RT1	RT0	0	0	PLVD	OPEN	Remote Temperature 3 (Low Byte) ⁽³⁾
21	00	NC7	NC6	NC5	NC4	NC3	NC2	NC1	NC0	N Correction 1
22	00	NC7	NC6	NC5	NC4	NC3	NC2	NC1	NC0	N Correction 2 ^{(2) (3)}
23	00	NC7	NC6	NC5	NC4	NC3	NC2	NC1	NC0	N Correction 3 ⁽³⁾
FC	X	X	X	X	X	X	X	X	X	Software Reset ⁽⁵⁾
FE	55	0	1	0	1	0	1	0	1	Manufacturer ID
FF	21	0	0	1	0	0	0	0	1	TMP421 Device ID
		0	0	1	0	0	0	1	0	TMP422 Device ID
		0	0	1	0	0	0	1	1	TMP423 Device ID

De la sección del modulo I2C ubicamos al componente en las líneas 2 del BMC (tambien esta conectado al PCH en diferente puerto), y observamos que tiene la dirección 0x98 para obtener información del IC necesitamos conocer los registros internos del mismo, buscamos en la hoja de datos y observamos que los registros 0x0100 0x0200 y 0x0300 tienen el dato que necesitamos, como cada dispositivo esclavo del bus es diferente es necesario revisar las hojas de datos para poder usarlos.

Probablemente el proveedor del BMC que estemos usando nos proveera de informacion sobre el uso del modulo I2C incluido en el IC, el cual puede verse en la figura anterior.

Imagen Ficticia de muestra com podria ser una implementacion del modulo I2C

```
10 mov SMBP, 0x98      ;cargamos la direccion en el registro de salida del SMBUS #2
11 mov SMBS, b'1010    ;indicamos que vamos a leer del dispositivo bit #4 los tres primeros bits
12          ;indican cual bus del BMC vamos a usar
13 mov cuenta,ECX     ;guardamos el valor de ECX antes de iniciar el ciclo
14 mov ECX,1           ;solo leeremos un byte
15 L1:
16 call SMBUS_PROC   ;con los datos en los registros llamamos a la rutina del
17 loop L1            ;BMC para que realice la lectura
18 mov AC, SMB2P       ;recuperamos el Byte que devuelve la rutina SMBUS2_PROC
```

Registros ficticios del BMC

SMBP puerto de salida

SMBS registro de configuracion

Funcion ficticia

SMBUS_PROC para manipular el
modulo SMBus del BMC

Esta imagen es Ficticia, no es una
implementacion Real, es meramente
ilustrativa

El proveedor de nuestro BMC (usualmente Aespeed), podria incluir en la hoja de datos alguna funcion pre-construida para usar los puertos del Modulo de I2C del IC, en caso de no contar con alguna tendriamos que implementarla nosotros, como es en el caso de muchos microcontroladores economicos.

El BMC tiene la posibilidad de cargar un sistema operativo pequeño para hacer mas faciles las operaciones de monitoreo del servidor



Podriamos obtener una respuesta como 0x1C igual a 28 grados, tambien existe la posibilidad de usar el comando en un script y realizar operaciones mas complejas, como leer los datos del servidor de la memoria EEPROM como: numero de serie, lote, direccion mac, fecha de manufactura, etc.

Podriamos obtener una salida mas detallada:

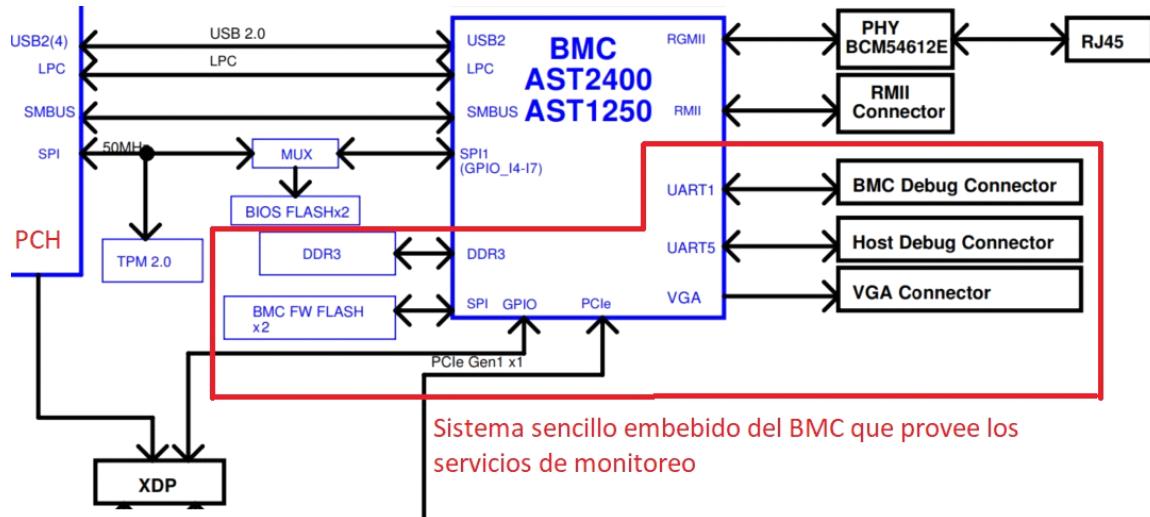
C:\>I2Cmac

MAC Adress:

{FC5CF426-4D12-4791-A34C-17263ED630A0}

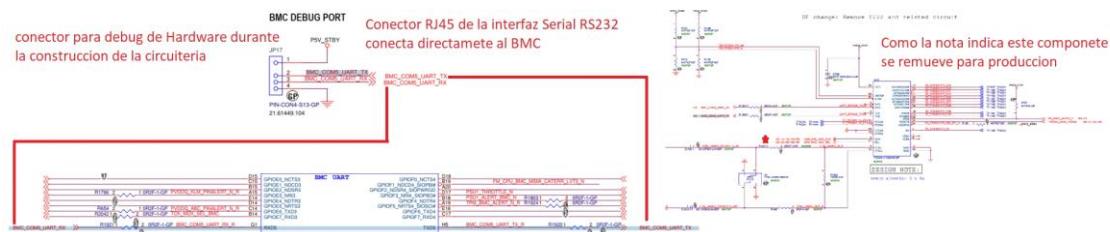
El comando podria incluir una sintaxis propia del cualquier otro comando del un interprete de ordenes.

Como mencionamos arriba, toda esta abstraccion es provista por una computadora embebida en el BMC, la cuia nos ofrece una plataforma bastante completa para monitorear , administrar y modificar el sistema, en pocas palabras dar "servicio de mantenimiento", cuenta con una interfaz sencilla serial de la cual se hablará mas adelante.



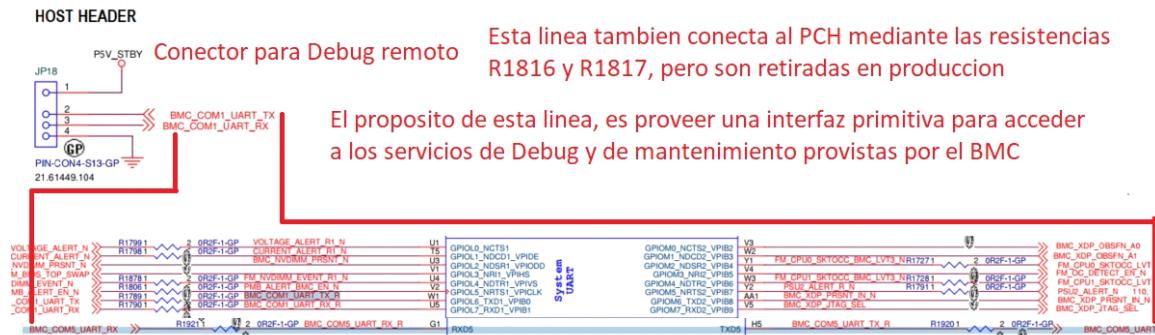
RS232 interfaz TTY

El sistema tiene dos líneas de interfaz TTY para el diagnóstico, esta primera linea se utiliza en las primeras etapas de diseño, como cuando se está implementando la funcionalidad del PCH y solo contamos con el BMC.



Esta segunda linea es la que usualmente se utiliza para dar servicio al servidor de manera remota, puede estar disponible o no al usuario final, dependiendo del fabricante, esta es la linea que conecta la interfaz

TTY con el sistema operativo del BMC (sistema embebido).



El funcionamiento de esta interfaz es bastante sencillo, proviene de las viejas terminales TTY que se usaron para transmitir datos mecanografiados durante la época del telegrafo, posteriormente dieron un salto con los sistemas multiusurio de las primeras computadoras, tambien formaron parte del origen del sistema ASCII, aunque es un sistema muy confiable tiene un ancho de banda limitado y es probable sea desplazado por alguna otra interfaz como la USB.

La siguiente figura representa como se implementa esta interfaz:

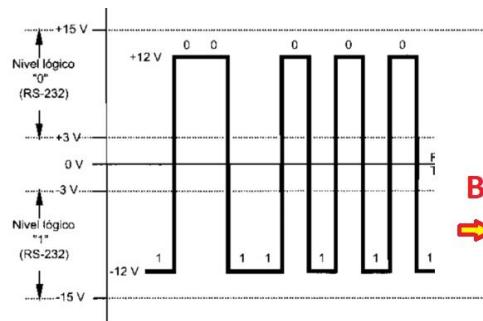


Figura 20-3 Niveles de tensión lógicos para RS232

Carácter	ASCII	Carácter	ASCII
A	0100 0001	W	0101 0111
B	0100 0010	X	0101 1000
C	0100 0011	Y	0101 1001
D	0100 0100	Z	0101 1010
E	0100 0101	0	0011 0000
F	0100 0110	1	0011 0001
G	0100 0111	2	0011 0010
H	0100 1000	3	0011 0011
I	0100 1001	4	0011 0100
J	0100 1010	5	0011 0101
K	0100 1011	6	0011 0110
L	0100 1100	7	0011 0111
M	0100 1101	8	0011 1000
N	0100 1110	9	0011 1001
O	0100 1111	+	0010 1011
P	0101 0000	-	0010 1101
Q	0101 0001	*	0010 1010
R	0101 0010	:	0011 1010
S	0101 0011	=	0011 1101
T	0101 0100	<	0011 1100
U	0101 0101	:	0011 1011
V	0101 0110		0011 1011

El código ASCII es representado con valores lógicos típicos de los componentes TTL (A), para posteriormente ser convertidos en valores RS232 (B),

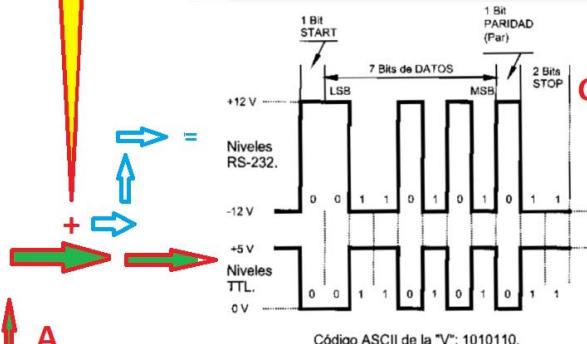


Figura 20-4 Ejemplo de envío de un byte según normas RS232

Basicamente el Código ASCII nos indica como se codifican las letras en binario, podemos ver como esta codificacion se implementa en logica TTL (como la que usamos en I2C), con la cual estamos mas familiarizados, posteriormente esta señal es convertida a una señal RS232, este tipo de señal pertenece al grupo de pares diferenciales, estas señales exhiben caracteristicas utilies para la transmicion a distancia, debido a que no cuentan con una linea de reloj a diferencia del I2C, la frecuencia de transmicion debe de conocerse de antemano, una medida usada para cuantificar la cantidad de informacion usada es el Baudio, definido como el numero de bits enviados por segundo.

Como los puertos RS232 no cuentan con linea de reloj existe de antemano una lista de "convenciones" de la velocidad de transmicion, entre las que encontramos 1200, 4800, 9600 115200, etc, los cuales son velocidades muy lentas, pero lo suficientemente rapidas para sostener una interfaz serie de servicio.

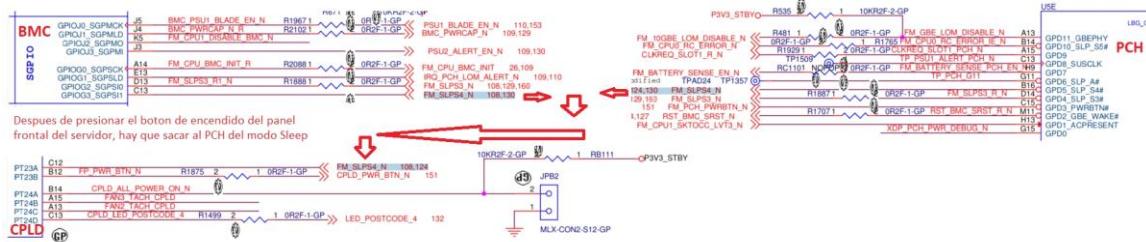
Existen mas servicios que puede proporcionar el BMC (ISCP, Etherneta, Video, SPI, AC-to-DC,etc), pero por el momento los dejaremos de lado para continuar con la descripcion de la secuencia de POWER ON del servidor, mas tarde regresaremos a explicar mas de ellos.

ETAPA MAIN

Tras presionar el boton de encendido del panel frontal, o enviar la señal de encendido por software, el servidor comenzara a encender al "sistema principal" compuesto por el PCH, los CPUs, RAM, etc.

FM_SLPS4_N

Lo primero es iniciar al PCH, que se encuentra en modo de bajo consumo "sleep", y avisar de este evento a los componentes implicados, para que por ejemplo el BMC se prepare para "ceder" el control al PCH y al CPLD para comenzar la secuencia de "bootedo" del sistema "mayor".



FM_SLPS2_N

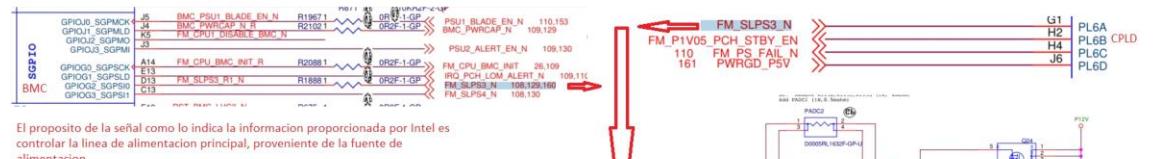


Table 2-24. Power Management Interface Signals (Sheet 3 of 4)

Name	Type	Description
SLP_S4#	0	R3 Sleep Control: SLP_S4# is for power down control. This signal shuts off power to all non-critical systems when in S3 (Suspend To RAM), S4 (Suspend To Disk), or S5 (Soft Off) states.



Comenzamos por activar la linea de alimentacion principal del servidor, volvemos a encontrar el Hotswap Controller otra vez, para mayor informacion sobre el funcionamiento del ADM1172 revise la hoja de datos.

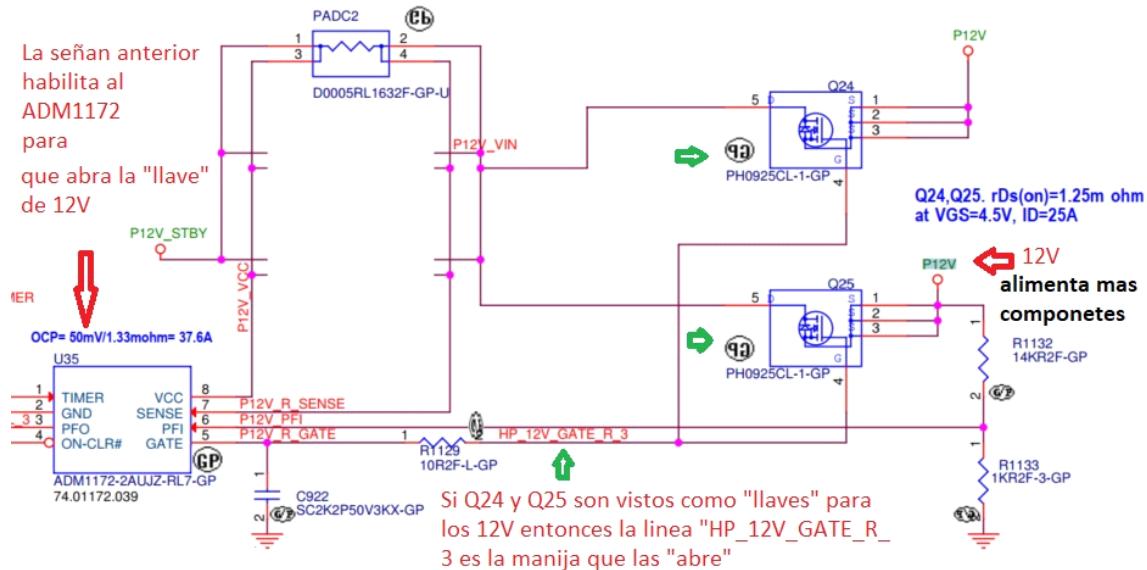
FM_PS_EN

Habilitar nuevamente al ADM1172, la señal puede parecer redundante, pero dado que el puerto del CPLD tambien puede ser entrada, podriamos usar esta linea para des-habilitar el hotswap



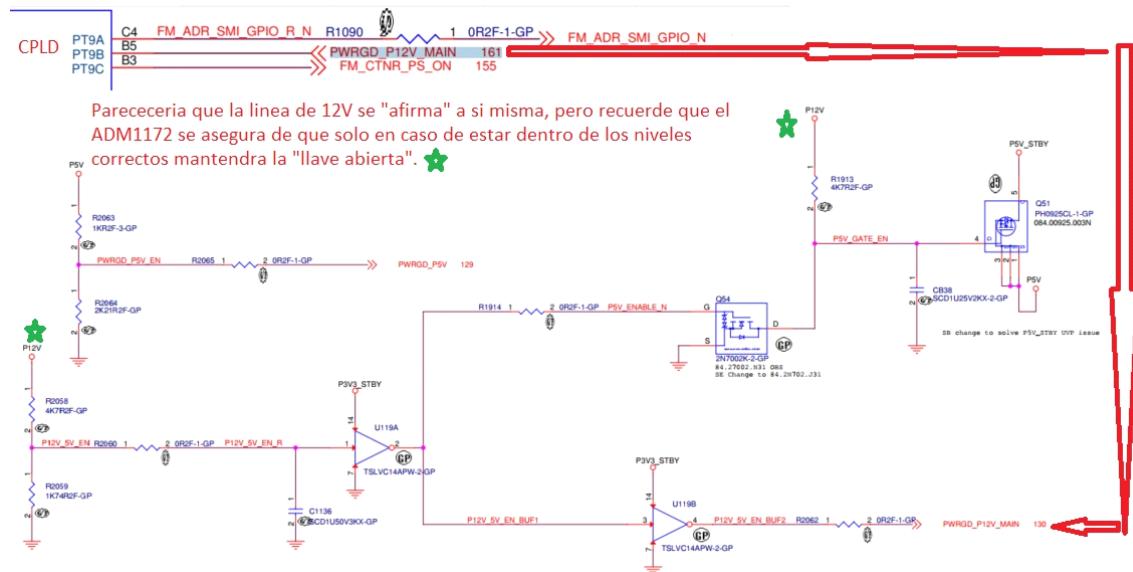
Esta señal parece redundante, pero dado que los puertos del CPLD pueden ser de entrada tambien, y como existe un "O" al inicio del puerto de ADMIN1172, debemos interpretar como "la desaparicion de la Señal", activa al dispositivo, esto puede ser util para poder realizar un Cool Reset, donde se retirara la alimentacion de las fuentes del sistema, en contraste con el Warm Reset.

P12V



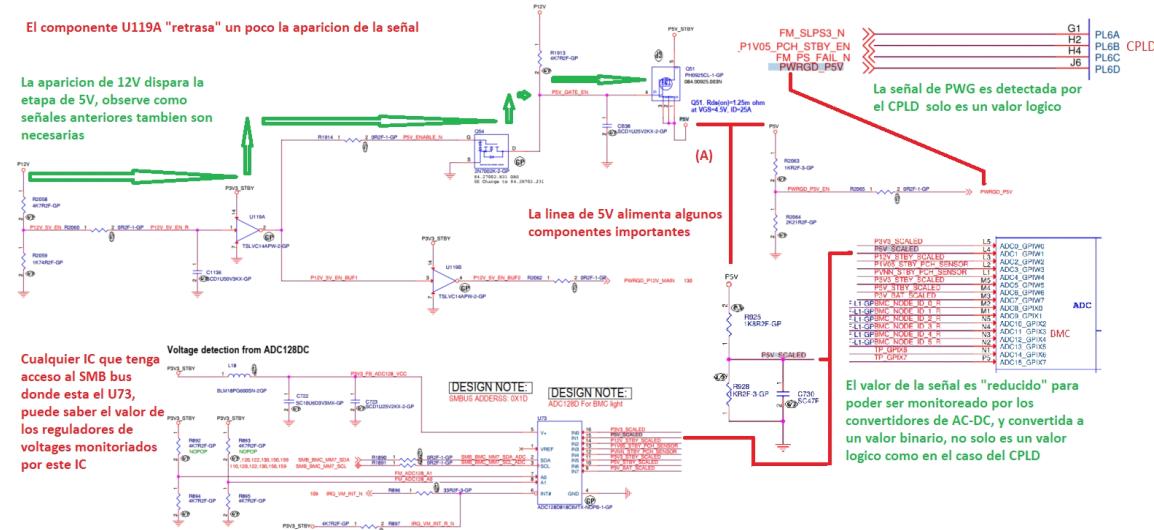
la señal FM_PS_EN y la siguiente, todas van "de la mano", el proposito es habilitar la linea de 12V de la etapa MAIN, que como observamos anteriormente con los 12V_STBY es una de las lineas principales de alimentacion.

PWRGD_P12V_MAIN



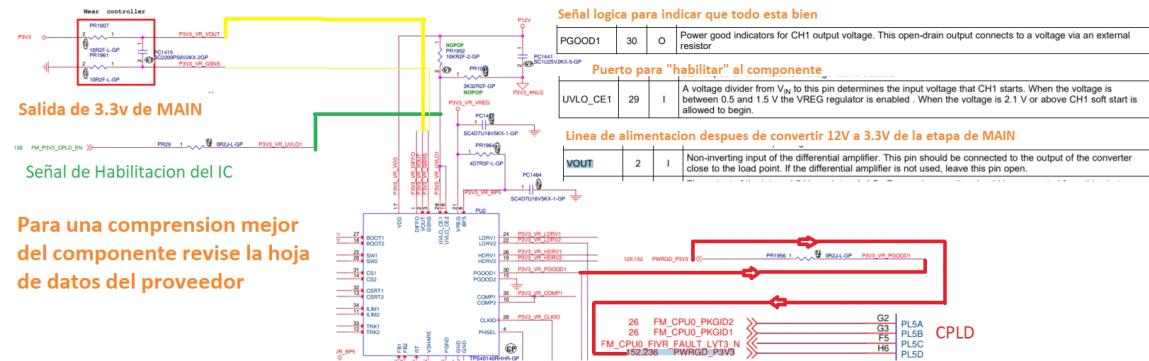
Las tres señales FM_PS_EN, P12V, PWRGD_P12V_MAIN giran entorno al IC U35 ADM 1172, el Hot Swap controller, para habilitar la linea principal de 12V de la etapa MAIN, estan muy relacionadas y cada una de ellas en consecuencia de la linea anterior, (lo cual se ha venido observando), merece mención porque todas convergen alrededor de un solo componente.

P5V



El componente U73 juega un papel importante, ya que monitorea varios niveles de voltage importantes y debido a que cuenta con una interfaz I2C, permite que cualquier componente Maestro conectado al SMBus pueda "enterarse" de los valores medidos aunque no cuente con la función de AC-DC (conversion analoga a digital).

P3V3



Esta etapa en la secuencia, es bastante intrincada, para poder entender mejor como fue implementada, revisar la hoja de datos del PU2 (TPS40140), la funcion principal del IC es manejar dos fuentes sincronizadas, por lo que "entregar" un voltage de salida de 3.3V a la etapa MAIN, es un "adicional" a su funcion principal.

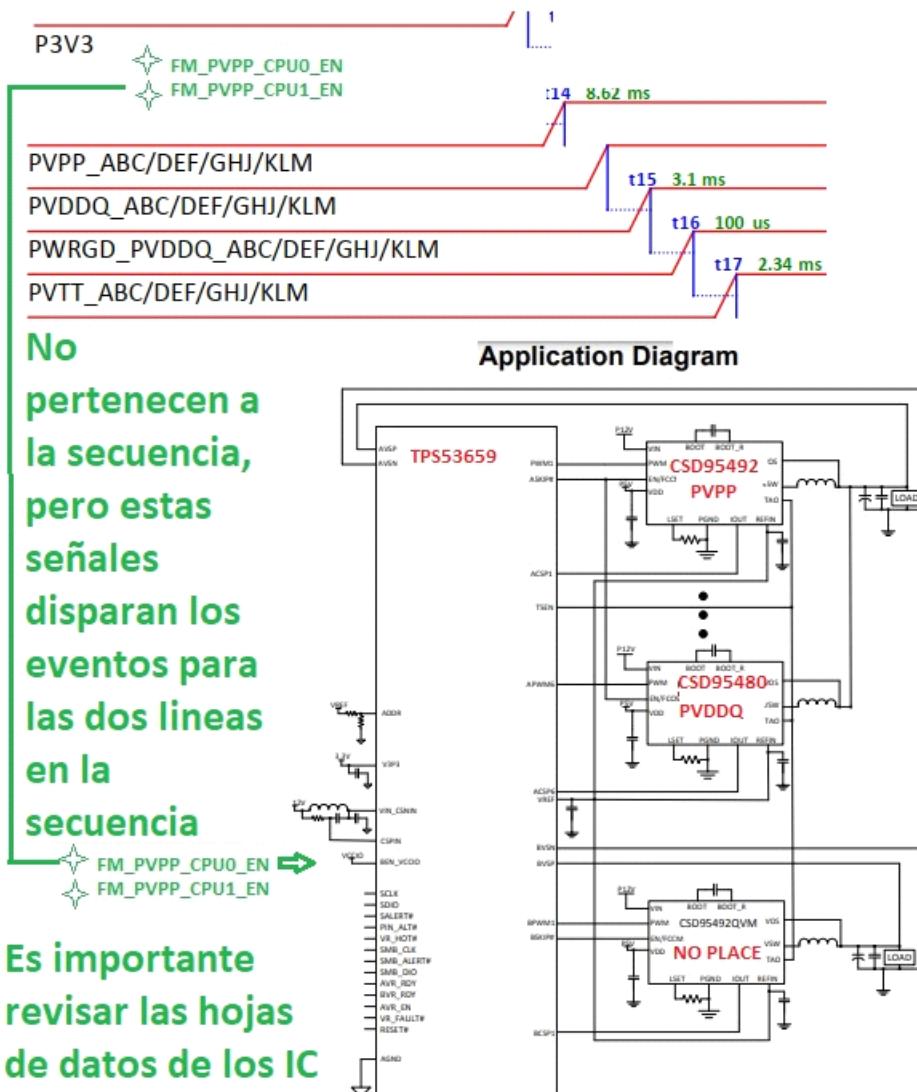
Hasta este punto en la etapa MAIN, se han alimentado (e inicializado), los componentes que se requieren para habilitar el sistema "Mayor", lo siguiente sera inicializar los CPUs y la memoria RAM

PVPP_ABC/DEF/GHJ/KLM y PVDDQ_ABC/DEF/GHJ/KLM

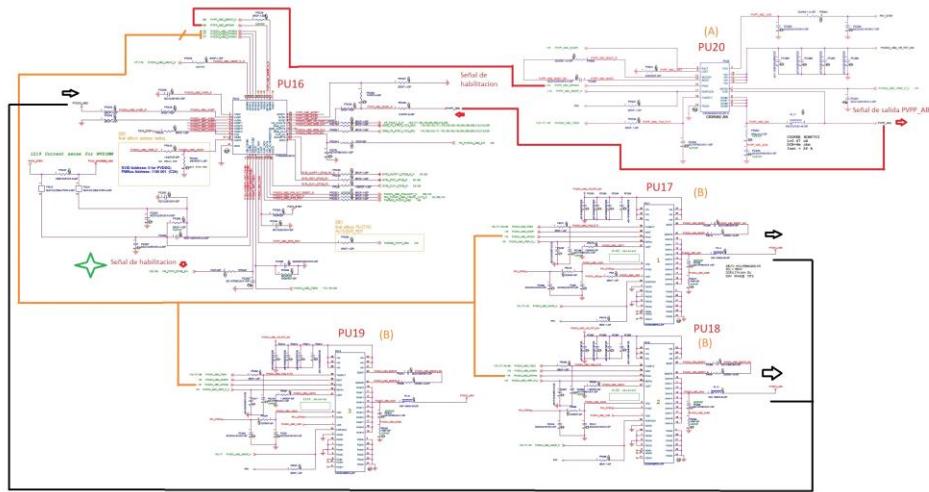
Son las siguientes cuatro líneas donde cada par pertenece a uno de los dos CPU

CPU 0	CPU 1
PVPP_ABC	PVPP_GHJ
PVPP_DEF	PVPP_KLM

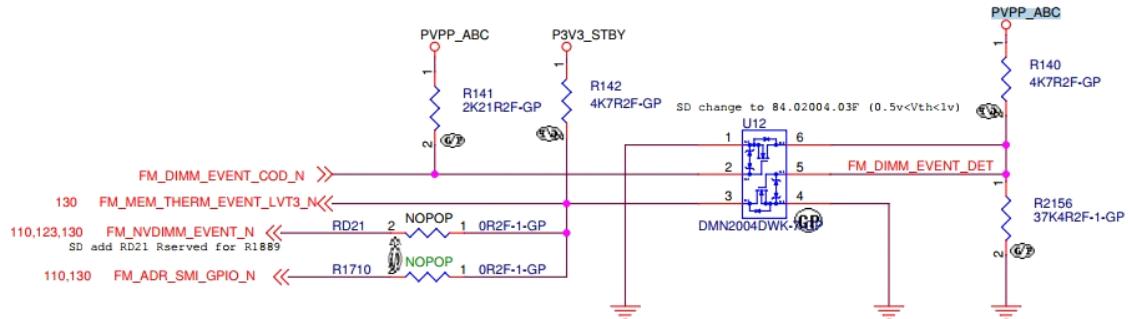
Las líneas giran entorno a los dispositivos TPS53659(NU16), CSD95492QVM (NU20) y los CSD95480(NU17-19) la hoja de datos de este componente indica lo siguiente (NU20):



Ya implementado en nuestro servidor:



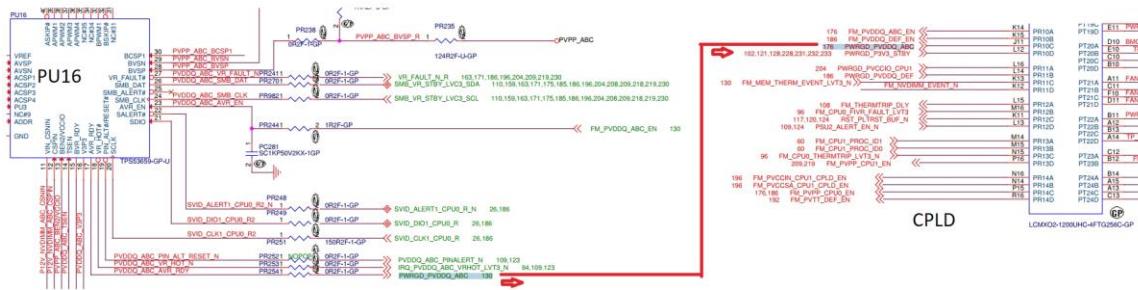
Lo que paraeece algo intrincado en realidad es una implementacion de fuentes de swicheo y un controlador de las mismas, debido a que la RAM consumira potencia acorde a la carga de trabajo que se le deposita en determinado momento, tambien nos interesa averiguar si los canales de RAM estan poblados:



Circuito para detectar la "presencia de los modulos de RAM" en este caso solo de los canales ABC

No tendria caso continuar con el proceso de Inicializacion sin no contamos con memoria RAM, nota VDD se refiere al voltaje referido para activar los dispositivos MOS y VDDQ es el voltaje para activar los puertos de entrada y salida de datos formados por transistores, VPP se refiere al voltaje requerido para programar ciertos tipos de memorias.

PWRGD_PVDDQ_ABC/DEF/GHJ/KLM



Tras que los dos voltajes anteriores se han estabilizado, esta señal indica que es posible pasar a la siguiente etapa en la secuencia de voltajes

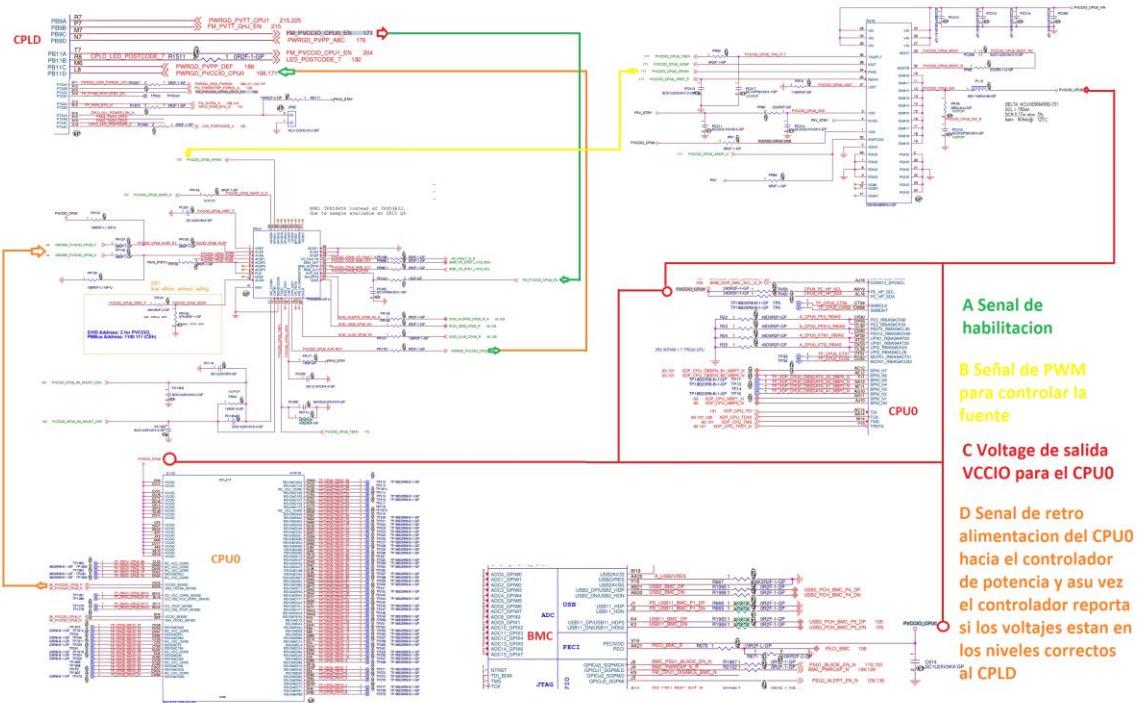
PVTT_ABC/DEF/GHJ/KLM



Todas estas señales son iniciadas por el CPLD, pero son consecuencia de la anterior etapa en la secuencia de power on, a diferencia del esquema "escalonado" de otras etapas de voltaje, ya que es requerido que el CPLD revise que entre etapas se efectue la inicialización de algunos componentes.

PVCCIO_CPU0/1

Comenzaremos a inicializar al CPU, la imagen esta enfocada para el CPU 0, pero es similar para el CPU1, (VCCIO se refiere al voltaje de los puertos de entrada y salida del CPU acorde a la hoja de datos)



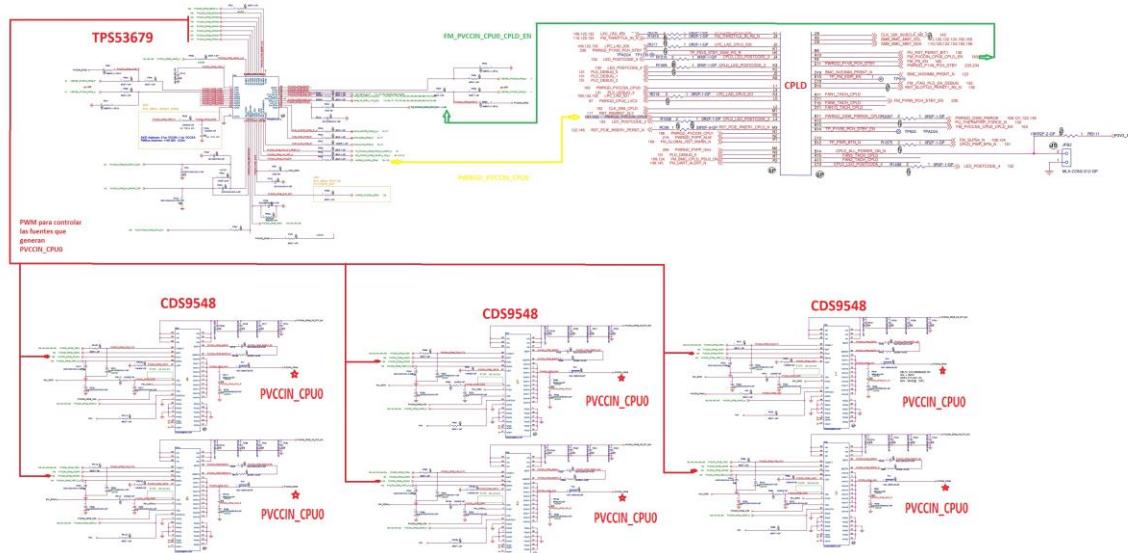
Como podemos apreciar los CPUs demandan un control con mas retroalimentacion debido a que el consumo de potencia puede variar, tras terminar la etapa PVTT_ABC/DEF/GHJ/KLM, el CPLD envia la señal FM_PVCCIO_CPU0_EN, para activar al controlador de la(s) fuente(s), mediante la aplicacion de una señal PWM, estas son las que generan el valor de alimentacion de la linea PVCCIO_CPU0/1, la salida es monitoreada por el CPU0/1 y estos reportan la informacion de nuevo al control de la(s) fuente(s) y este a su vez envia la señal de PWRGD al CPLD para pasar a la siguiente etapa.

FM_PVCCI_CPU0/1_CPLD_EN FM_PVCCIN_CPU0

PVCCIN_CPU0/1

PWRGD_PVCCIN_CPU0/1

Estas tres etapas estan relacionadas en la siguiente figura:

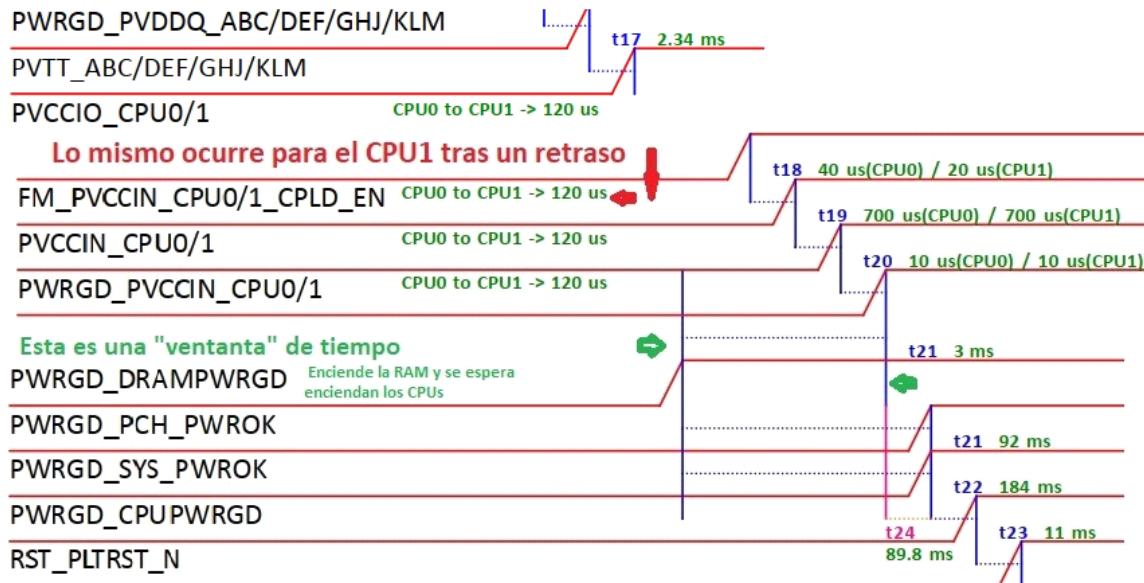


PVCCIN, es el voltaje de entrada primario para los reguladores dentro del CPU, es decir la alimentacion, no confundir con el voltaje de alimentacion de los puertos de entrada/salida.

Siempre es importante revisar la documentacion del componente para aclarar dudas:

V _{CCIN}	Primary voltage input to the voltage regulators integrated into the processor.
V _{SS}	Processor ground
V _{SSA}	System agent supply for Intel UPI and PCIe
V _{CCIO}	IO voltage supply input
V _{CCD}	DDR power rail
x1, x4, x8, x16	Refers to a Link or Port with one, two, four or eight Physical Lane(s)

Al final en estas tres etapas lo que se pretende es encender al CPU0, tomando en cuenta el diagrama de tiempos de la secuencia de POWER ON:



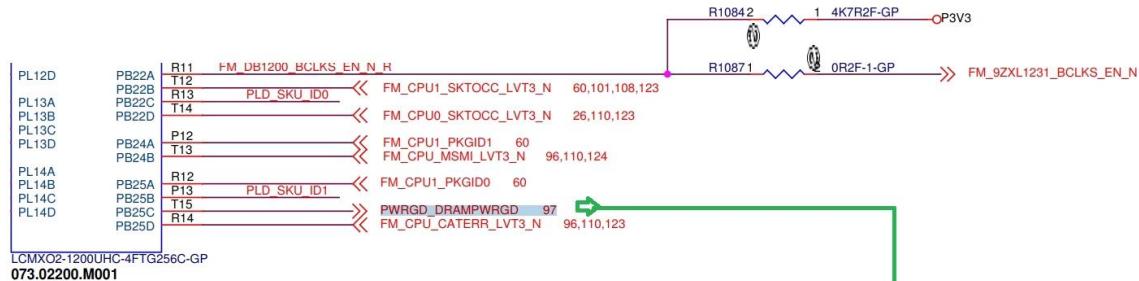
PWRGD_DRAMPWRGD

Revisando el diagra de tiempos (figura superior), podemos ver que la señal ocurre antes de encender los CPU e Inmediatamente despues de encender la memoria RAM.

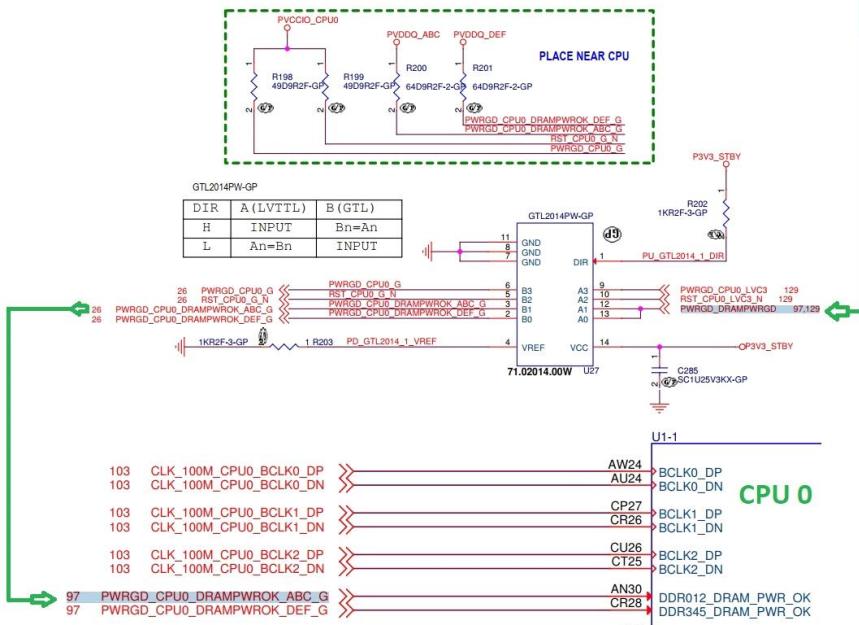
Table 4-2. Memory Channel Miscellaneous

Signal Name	Description
DDR {012,345}_RESET_N	System memory reset: Reset signal from processor to DRAM devices on the DIMMs. DDR012_RESET_N is used for memory channels 0, 1 and 2 while DDR345_RESET_N is used for memory channels 3, 4 and 5.
DDR{012,345}_SPDSCL	SMBus clock for the dedicated interface to the serial presence detect (SPD) and thermal sensors (TSoD) on the DIMMs. DDR_SCL_C012 is used for memory channels 0, 1 and 2 while DDR_SCL_C345 is used for memory channels 3, 4 and 5.
DDR{012,345}_SPDSA	SMBus data for the dedicated interface to the serial presence detect (SPD) and thermal sensors (TSoD) on the DIMMs. DDR_SDA_C012 is used for memory channels 0, 1 and 2 while DDR_SDA_C345 is used for memory channels 3, 4 and 5.
DDR{5:0}_CAVREF	DIMM Command address VREF signal
DDR{012,345}_DRAM_PWR_OK	Power good for VCCD rail used by the DRAM. This is an input signal used to indicate the VCCD power supply is stable for memory channels 0, 1, 2 and channels 3, 4, 5.
DDR{012,345}_RCOMP[2:0]	DDR Compensation resistance control

El propósito de esta señal es indicar que la memoria RAM está lista, y dar tiempo al CPU para que se inicialice, se puede ver en la figura siguiente que la señal debe pasar por un traductor de voltaje.



CPU_Resets_Power_Good



PWRGD_PCH_PWROK

Acorde a la tabla de la hoja de datos del PCH:

2.21 Power Management Interface

Table 2-22. Power Management Interface Signals (Sheet 1 of 3)

Name	Type	Description
PLTRST#	O	<p>Platform Reset: The PCH asserts PLTRST# to reset devices on the platform (such as SIO, FWH, LAN, processor, etc.). The PCH asserts PLTRST# during power-up and when S/W initiates a hard reset sequence through the Reset Control register (I/O port CF9h). The PCH drives PLTRST# active a minimum of 1 ms when initiated through the Reset Control register (I/O port CF9h).</p> <p>Notes:</p> <ol style="list-style-type: none"> 1. PLTRST# is in the VccSus3_3 well. 2. All CPU PCIe slots and devices should be connected to an independent signal (CPU PCIe RST#) derived from a BIOS controllable GPIO. The reset for the PCH PCIe devices can optionally be derived from another GPIO. The PCH asserts PLTRST# during power-up and when a hard reset sequence is initiated through the CF9h register. The dedicated PCIe reset GPIO(s) should also assert with PLTRST# and independently through BIOS control.
PLTRST_PROC#	O	Platform Reset Processor: A 1.0 V copy of PLTRST# pin. This signal is the main host platform reset and should directly connect to the processor pin PLTRSTIN#. No on-board logic is required to level shift the voltage of this signal.
PWRBTN#	I	<p>Power Button: The Power Button will cause SMI# or SCI to indicate a system request to go to a sleep state. If the system is already in a sleep state, this signal will cause a wake event. If PWRBTN# is pressed for more than 4 seconds, this will cause an unconditional transition (power button override) to the S5 state. Override will occur even if the system is in the S1-S4 states. This signal has an internal pull-up resistor and has an internal 16 ms de-bounce on the input. This signal is in the DSW well.</p>
PCH_PWROK	I	<p>Power OK: When asserted, PCH_PWROK is an indication to the PCH that all of its core power rails have been stable for at least 10 ms. PCH_PWROK can be driven asynchronously. When PWROK is negated, the PCH asserts PLTRST#.</p> <p>Notes:</p> <ol style="list-style-type: none"> 1. It is required that the power rails associated with PCI/PCIe (typically the 3.3 V, 5 V, and 12 V core well rails) have been valid for 99 ms prior to PWROK assertion in order to comply with the 100 ms PCI 2.3/PCIe* 2.0 specification on PLTRST# de-assertion. 2. PCH_PWROK must not glitch, even if RSMRST# is low.
RI#	I	Ring Indicate: This signal is an input from a modem. It can be enabled as a wake event, and this is preserved across power failures.
RSMRST#	I	Resume Well Reset: This signal is used for resetting the resume power plane logic. This signal must be asserted for at least t201 after the suspend power wells are valid. When de-asserted, this signal is an indication that the suspend power wells are stable.

Una
 consecuencia
 de la
 aparicion de
 esta señal es
 iniciar el
 proceso de
 carga del
 BIOS como se
 indica en la
 tabla inferior

Table 2-16. Functional Strap Definitions (Sheet 2 of 4)

Signal	Usage	When Sampled	Comment															
GPIO51	Boot BIOS Strap bit 1 (BBS1)	Rising edge of PCH_PWROK	<p>This signal has a weak internal pull-up. This field determines the destination of accesses to the BIOS memory range. Also controllable using Boot BIOS Destination bit (Chipset Config Registers: Offset 3410h:Bit 11). This strap is used in conjunction with Boot BIOS Destination Selection 0 strap.</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th>Bit11</th> <th>Bit 10</th> <th>Boot BIOS Destination</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>1</td> <td>Reserved</td> </tr> <tr> <td>1</td> <td>0</td> <td>Reserved</td> </tr> <tr> <td>1</td> <td>1</td> <td>SPI (default)</td> </tr> <tr> <td>0</td> <td>0</td> <td>LPC</td> </tr> </tbody> </table> <p>Notes:</p> <ol style="list-style-type: none"> 1. The internal pull-up is disabled after PLTRST# de-asserts. 2. If option 00 (LPC) is selected, BIOS may still be placed on LPC, but the platforms is required to have SPI flash connected directly to the PCH's SPI bus with a valid descriptor in order to boot. 3. Boot BIOS Destination Select to LPC/PCI by functional strap or using Boot BIOS Destination Bit will not affect SPI accesses initiated by Intel® ME or Integrated GbE LAN. 4. See Chapter 10 for additional information. 5. This signal is in the Core well. 	Bit11	Bit 10	Boot BIOS Destination	0	1	Reserved	1	0	Reserved	1	1	SPI (default)	0	0	LPC
Bit11	Bit 10	Boot BIOS Destination																
0	1	Reserved																
1	0	Reserved																
1	1	SPI (default)																
0	0	LPC																

Basados en el diagrama de tiempos de la secuencia de POWER ON y la informacion de esta tabla, podemos deducir que tras "anuciar" que el el CPU se ha enviado a encender, debemos esperar por lo menos 10 ms para verificar que todos los rieles de voltge estan operando correctamente y entonces el PCH comenzara a realizar algunas tareas como la "carga" del BIOS.

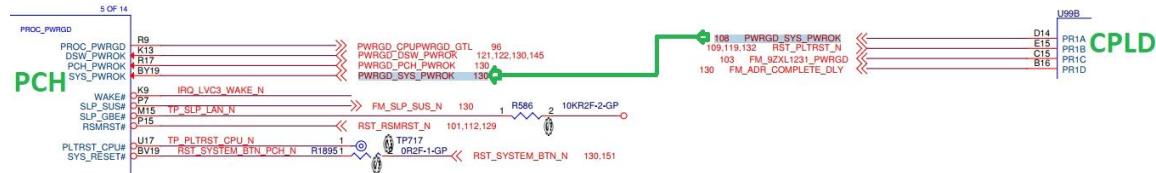
PWRGD_SYS_PWROK

La siguiente tabla (obtenida de la hoja de datos) indica la funcion de esta señal:

Table 2-24. Power Management Interface Signals (Sheet 3 of 4)

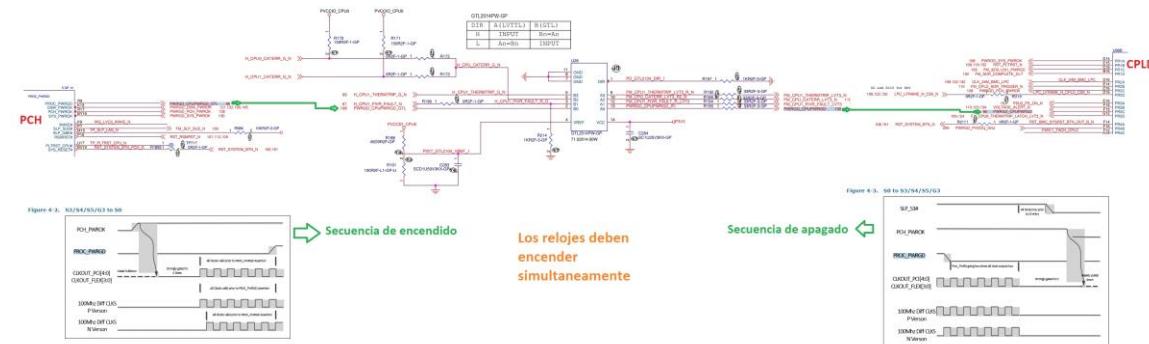
Name	Type	Description
SYS_PWROK	I	System Power OK: This generic power good input to the PCH is driven and utilized in a platform-specific manner. While PWRGD always indicates that the core wells of the PCH are stable, SYS_PWROK is used to inform the PCH that power is stable to some other system component(s) and the system is ready to start the exit from reset.

Entre esta señal y la anterior podemos inicializar rieles de alimentacion para dispositivos como PCI, por ejemplo.



PWRGD_CPUPWRGD

La siguiente imagen muestra como aunque una señal no parece clara si revisamos y seguimos el esquematico esta adquiere sentido:



La señal pasa por un traductor de voltajes y se combierte en PWRGD_CPUPWRGD_GTL, llegando al PCH esta conectada al puerto etiquetado como PROC_PWRGD, esta señal se refiere al inicio correcto de los "relojes" que dan vida al sistema, como indican los diagramas de tiempo (extraidos de la hoja de datos).

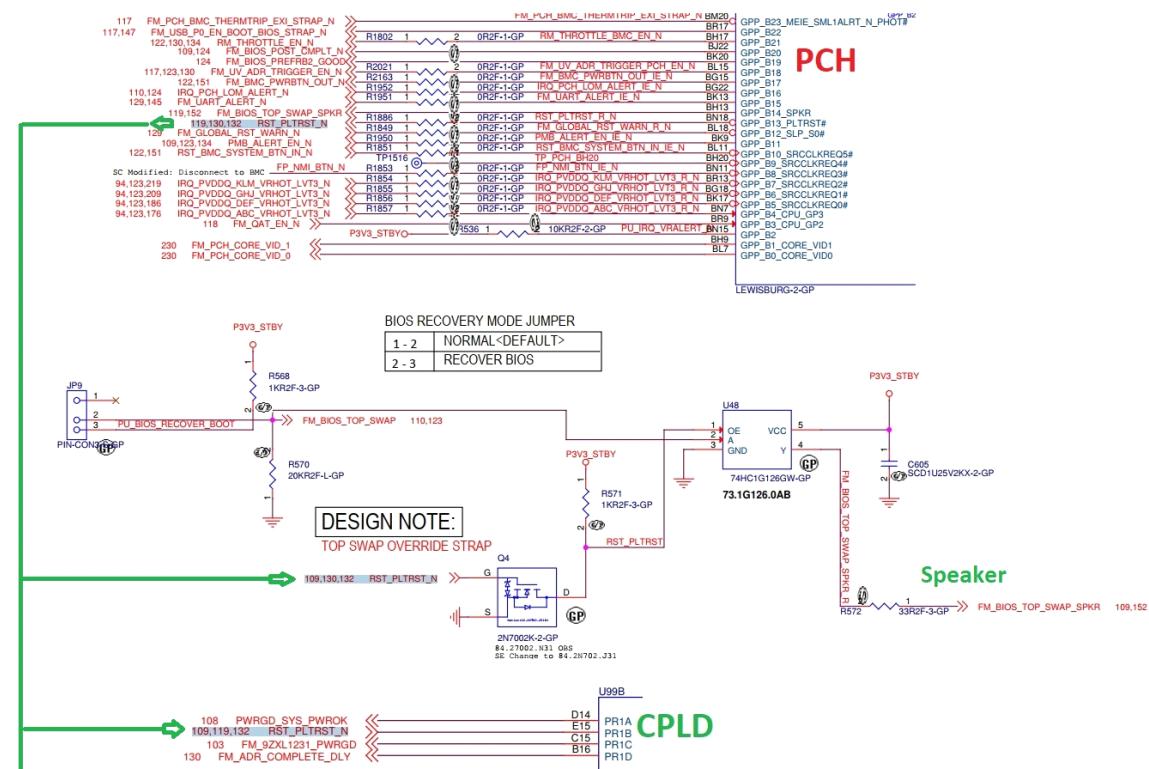
RST_PLTRST_N

La hoja de datos dice lo siguiente:

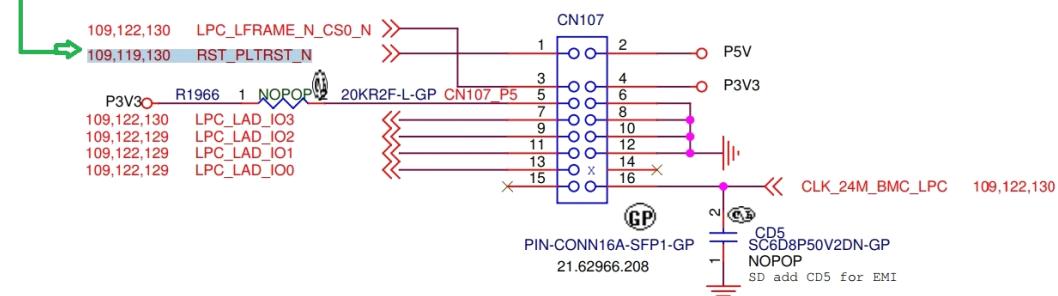
Table 2-24. Power Management Interface Signals (Sheet 2 of 4)

Name	Type	Description
PLTRST#	O	<p>Platform Reset: The PCH asserts PLTRST# to reset devices on the platform (such as SIO, FWH, LAN, processor, and so on). The PCH asserts PLTRST# during power-up and when S/W initiates a hard reset sequence through the Reset Control register (I/O port CF9h). The PCH drives PLTRST# active a minimum of 1 ms when initiated through the Reset Control register (I/O port CF9h).</p> <p>Note: PLTRST# is in the VccSUS3_3 well.</p>

Y el esquematico indica:



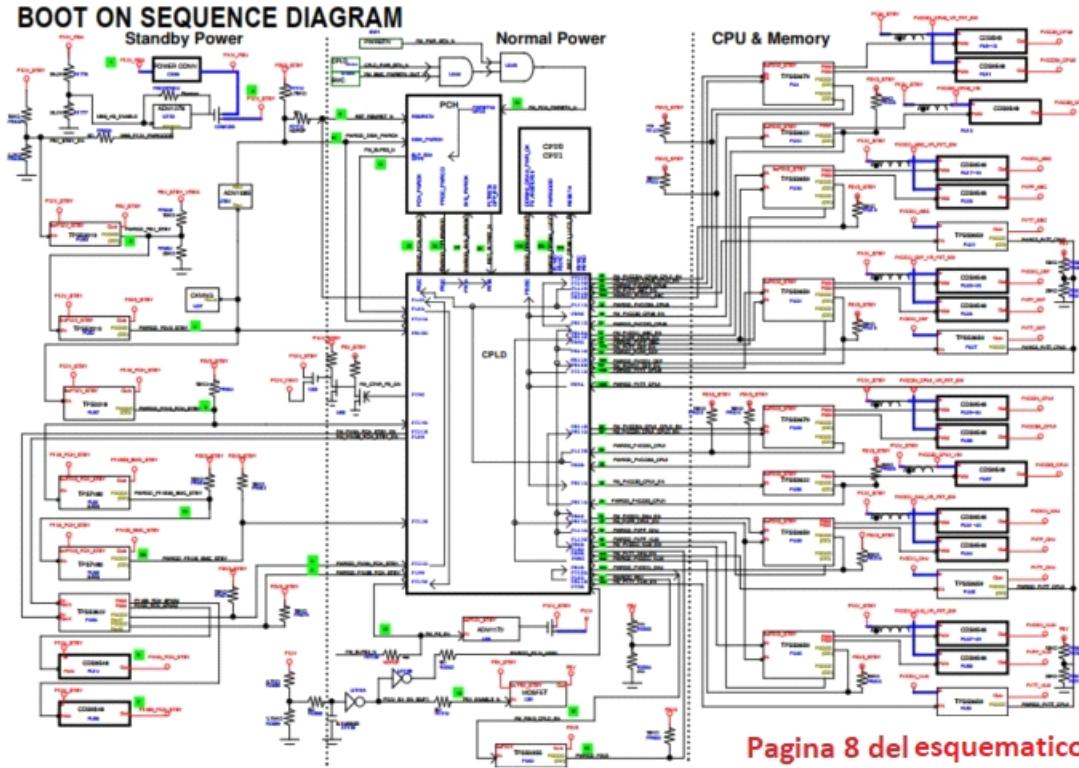
LPC debug header



Parece ser que la señal esta señal es el "gatillo"(Trigger) para un reset por Hardware, listo para cuando se presione el boton de Reset o se envie la señal por Software.

BOOT ON SEQUENCE

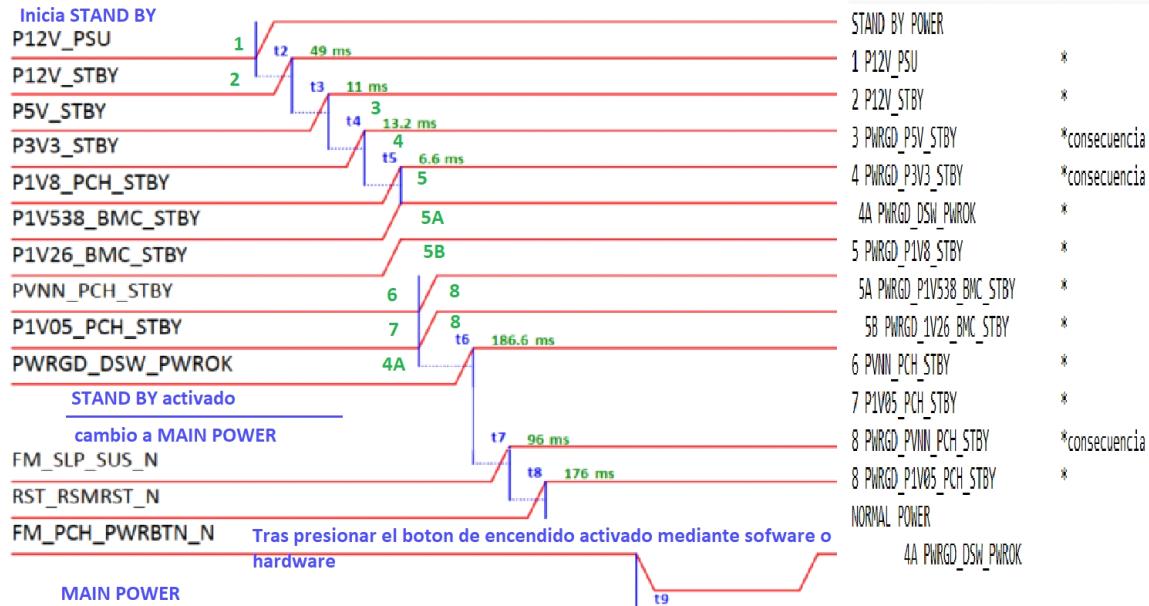
Repararemos la secuencia de encendido de un servidor, pero vista desde el punto de vista de la inicialización de los componentes principales.



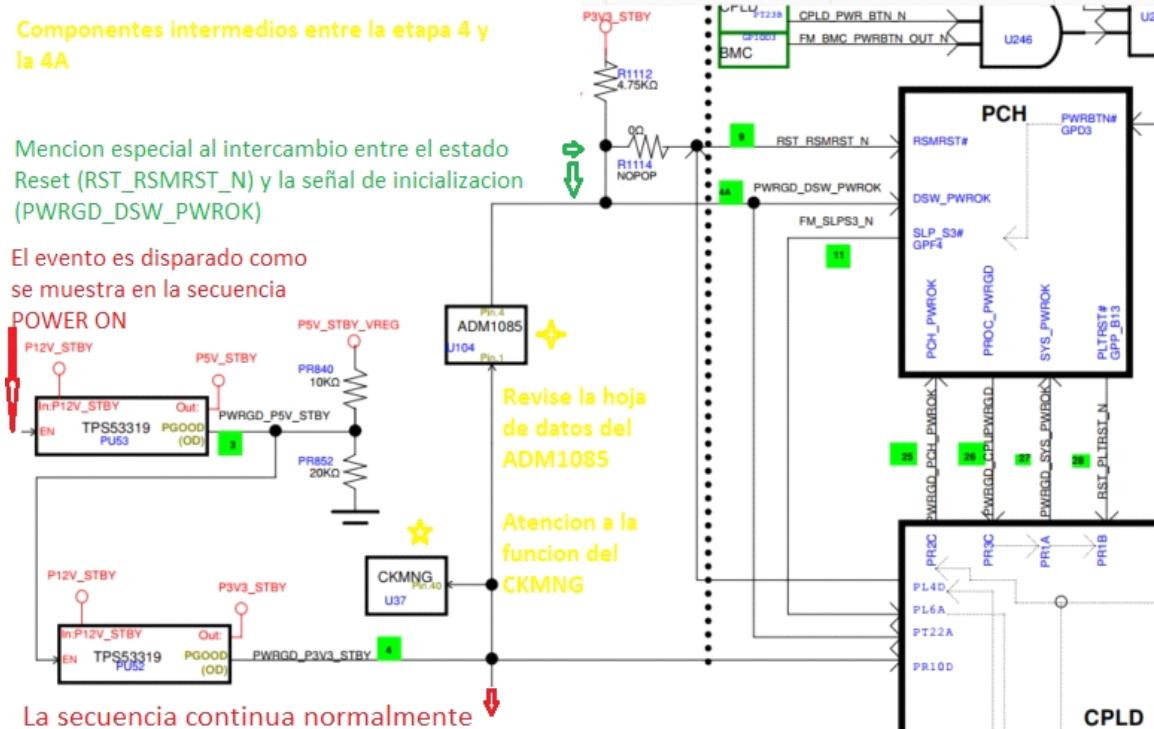
Página 8 del esquemático

Nos centraremos en los eventos que se refieren a la inicialización de componentes y que no han sido explicados ya en la secuencia de POWER ON (pagina 9 del esquemático) o que no aparecen en la misma.

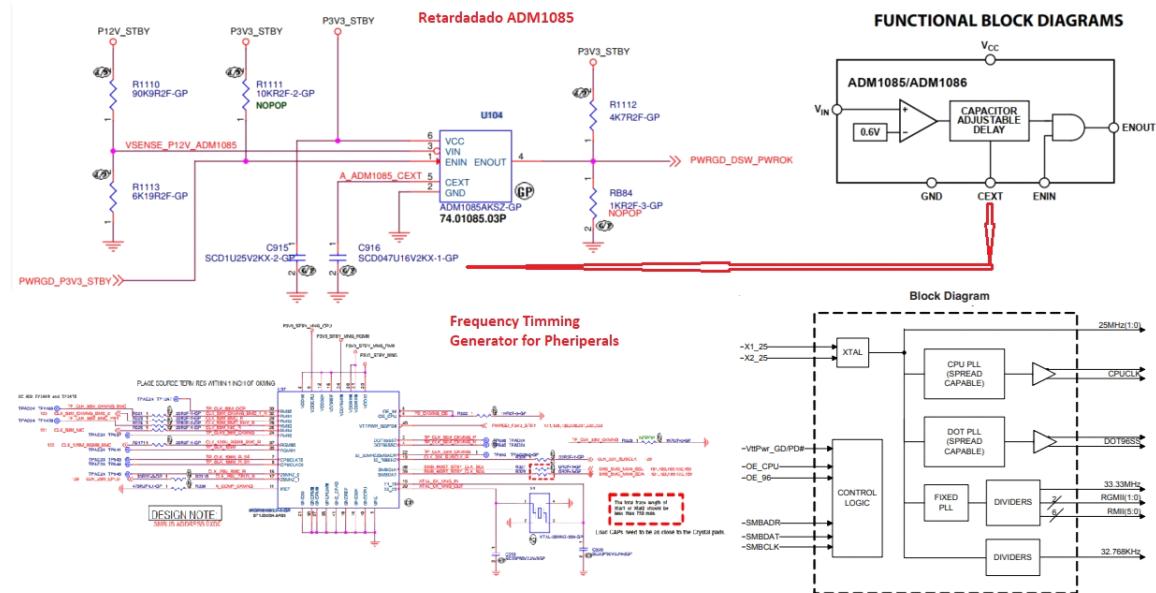
La parte izquierda del diagrama se refiere en su mayoría a eventos ya explicados en la secuencia de POWER ON, como podemos apreciar en la siguiente figura, donde colocamos a la derecha del diagrama de tiempos las señales en el orden listado por BOOT ON SEQUENCE:



Hay una mención especial entre la ETAPA 4 y la ETAPA 4A, la cual se explica mirando el esquemático de BOOT SEQUENCE con mas detalle:

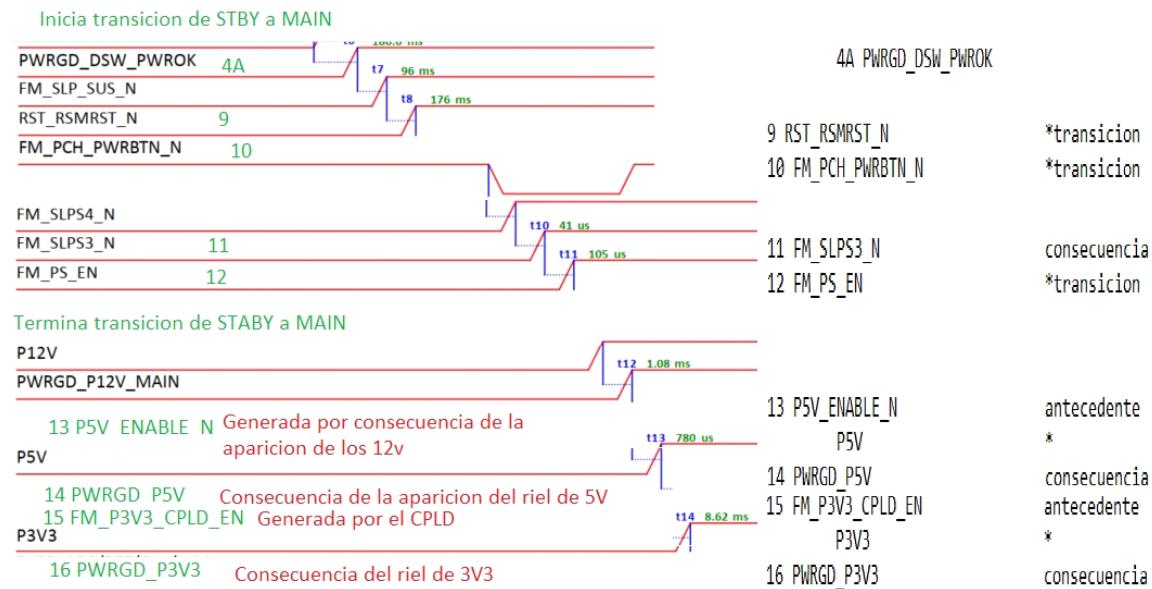


Mirando la hoja de datos de los componentes U37 y U104:



La idea es la siguiente, Tras alimentar al PCH hay que dejarlo un tiempo en el estado de RESET para evitar condiciones de **metaestabilidad**, posteriormente el PCH estara listo para iniciar la secuencia de MAIN inmediatamente tras la aparicion de la linea de P3V3_STBY, pero el ADM1085 retardara esto hasta despues de la linea P1V05_PCH_STBY este presente, dando tiempo para que varios perifericos necesarios puedan inicializarse y estar presentes.

Despues de entrar al "estado estable" STAND BY, estamos listos para hacer la transicion a MAIN y comenzar el inicio de los componentes, en este caso nos centramos en la "inicializacion" del CPLD

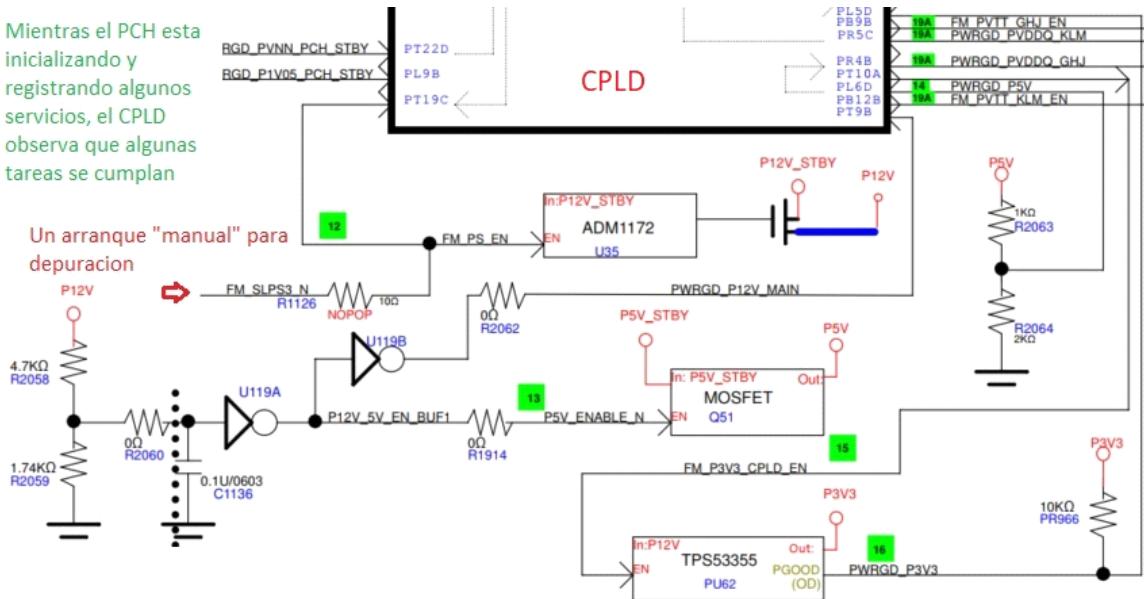


El CPLD es "el pegamento logico" que nos ayuda a realizar una transicion entre la etapa STAND BY y MAIN, cubre los "huecos" en la secuencia que no son cubiertos por el PCH o el CPU, aqui es donde

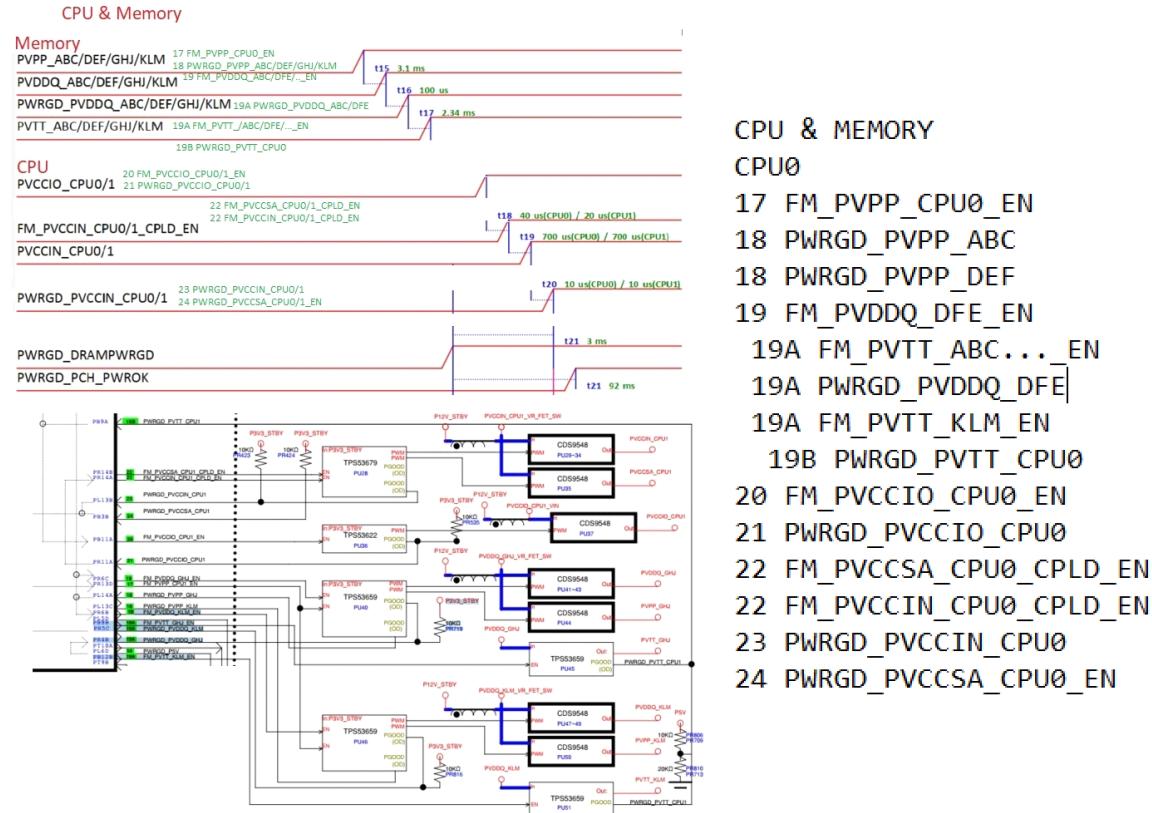
verdaderamente el diseñador tiene espacio para modificar el comportamiento de la secuencia de inicio del servidor.

Mientras el PCH esta inicializando y registrando algunos servicios, el CPLD observa que algunas tareas se cumplen

Un arranque "manual" para depuración



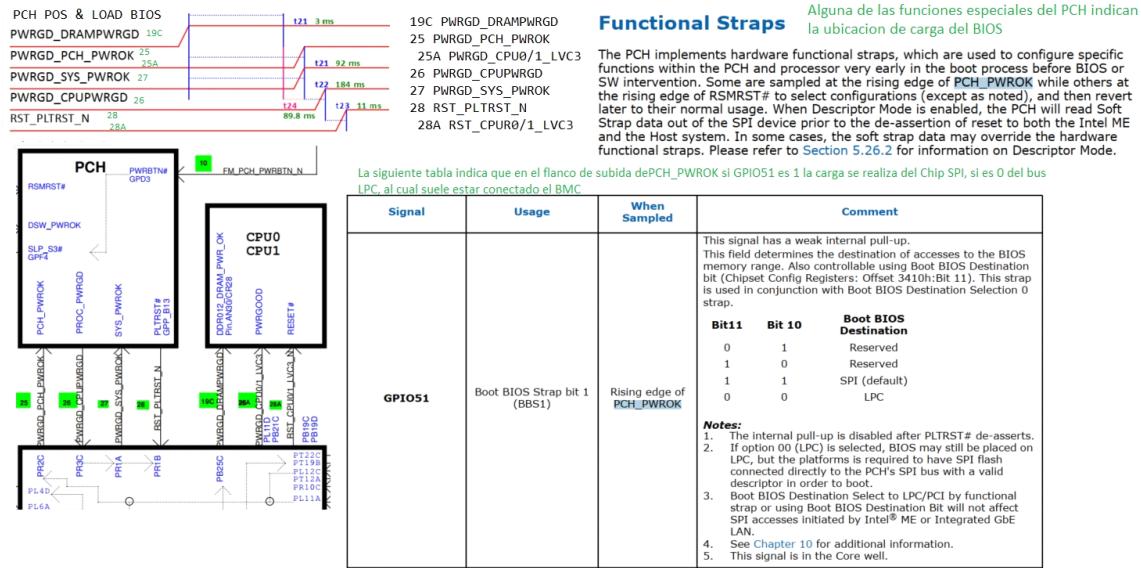
Como mencionamos anteriormente el diagrama de tiempos de la secuencia de POWER ON, ahora procederemos a inicializar formalmente al sistema "mayor" en el servidor, comenzando con la RAM y siguiendo con los CPUs.



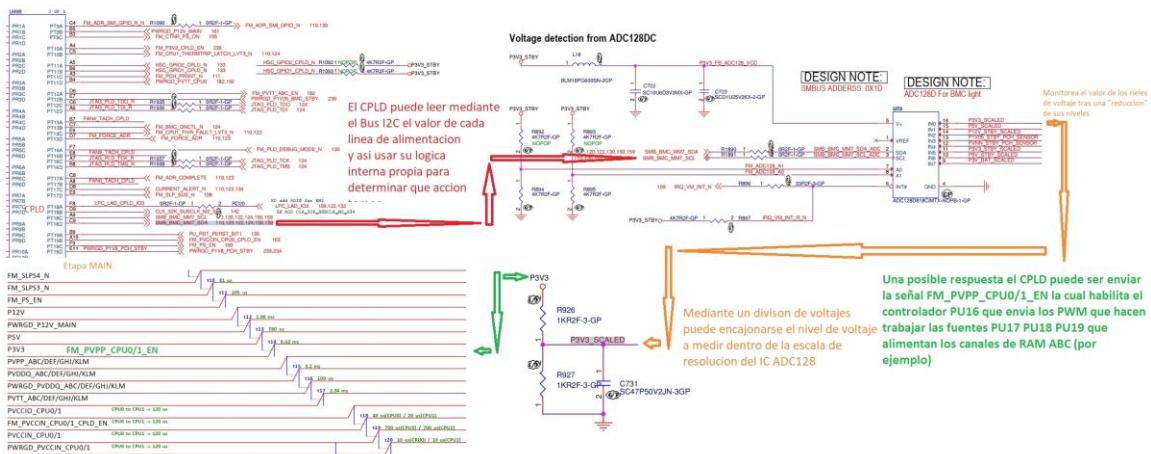
Una explicacion mas detallada se dio con anterioridad, se hace mencion aqui nada mas el enfasis en las señales logicas (color verde) en la imagen superior, las cuales "disparan" los eventos en la secuencia de POWER ON y son proporcionados por el CPLD.

PCH POS & LOAD BIOS

Tras que la memoria ha sido inicializada y los CPU ya estan encendidos, es posible entonces comenzar con la carga de BIOS y ejecutar el POS:

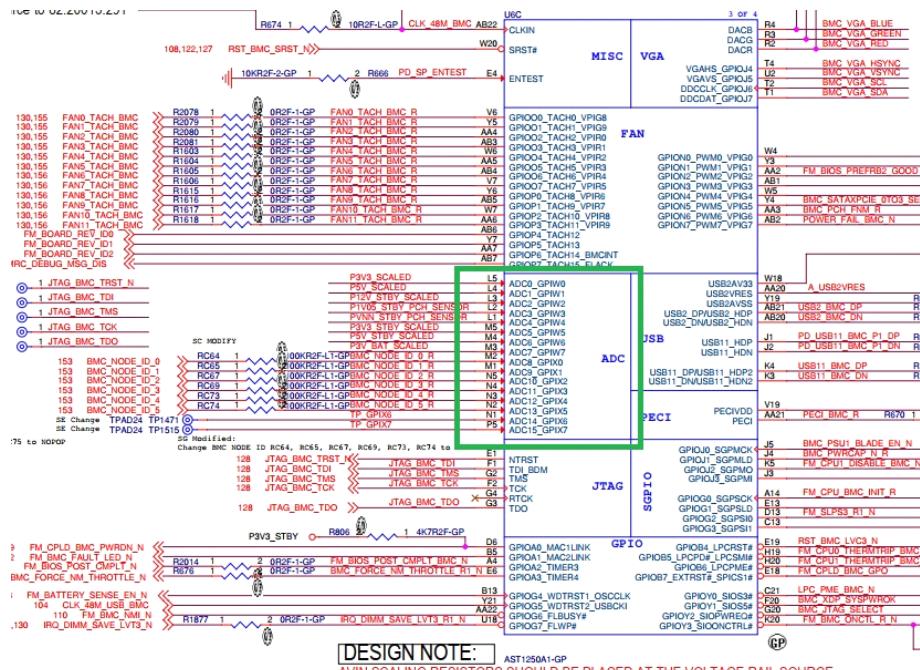


Una mencion a notar es que la etapa 26 PWRGD_CPUPWRGD y 27 PWRGD_SYS_PWROK de la BOOT SEQUENCE, respecto a POWER ON SEQUENCE es que estan en orden inverso, como vimos anteriormente SYS_PWROK es utilizado para indicar cuando algunos otros dispositivos (no PCH no CPU o RAM) han terminado de inicializarse, la secuencia de BOOT nos indica que primero se detecta que el CPU esta operando y despues que los demas dispositivos esten listos, en cambio en la secuencia de POWER ON es lo contrario, esto no afecta de ninguna manera el funcionamiento del servidor pero nos pone a pensar en cuando usar una u otra configuracion y cual seria la ventaja entre ellas.



Por ejemplo la secuencia podria ser dirigida por el CPLD, aunque este no cuente con convertidores ADC puede conectarse a una dispositivo que si lo tenga, tal como el ADC1280 que se muestra el cual monitorea los valores de las lineas de power on y puede comunicar el estado de las mismas mediante un bus I2C, de esa manera el CPLD podria enviar las señales de "habilitacion" para las fuentes de las diferentes etapas.

Tambien el BMC podria usarse para controlar las fuentes en la secuencia, ya que cuenta con un modulo de ADC y varios puertos GPIO (General Purpose Input/Output, Entrada/Salida de Propósito General), y de esta manera controlar la secuencia de POWER ON.



El BMC tambien cuenta con un modulo ADC, y posee muchos puertos GPIO, podriamos usar algunos disponibles para enviar las señales de Habilitacion a las fuentes o comunicarnos con el CPLD o algun otro dispositivo como un Microcontrolador para controlar las fuentes

En conclusion para el PCH y el CPU, mientras se cumplan las condiciones de arranque estos trabajaran sin problema independientemente de como se logre cumplir con estas condiciones.