

[Home](#)[Forums](#)[Resources](#)[Members](#)

```

skip 2nd initialization step
lv2(0): total memory size: 249MB+640KB
lv2(0): kern memory size: 12MB+640KB (heap:3492KB page pool:4736KB)
lv2(0): user memory size: 237MB
lv2(2):
lv2(2): Cell OS Lv-2 32 bit version 4.8.2
lv2(2): Copyright 2011 Sony Computer Entertainment Inc.
lv2(2): All Rights Reserved.
lv2(2):
lv2(2): revision: 50677
lv2(2): build date: 2017/08/24 13:19:54
lv2(2): processor: Broadband Engine Ver 0x0000 Rev 0x0202
lv2(2): PPU:0, Thread:0 is enabled.
lv2(2): PPU:0, Thread:1 is enabled.

```

At the same time, SYSCON UART:

```

Code:
>$ br
bringup
[SSM] state: 0000 -> 0101
Bringup Mode #0 (0xFF)
[SSM] ssmCb_OnStartingBePowOn() called.
[SSM] Bringup mode : syspm_stat=00000000/00000000
[POWSEQ] PowerSeq_Setup called.
[SSM] state: 0101 -> 0201
[POWSEQ] AV Backend Setup
[SSM] state: 0201 -> 0102
[SSM] state: 0102 -> 0202
[SSM] state: 0202 -> 0103
[SSM] state: 0103 -> 0203
[SSM] ssmCb_BeforeBeOn() called.
[SSM] state: 0203 -> 0104
Psbd_SbTransMode_Half:0x21e2
>$ 
[SSM] state: 0104 -> 0204
[SSM] state: 0204 -> 0105
[SSM] state: 0105 -> 0400
(PowerOn State)
[SERV NVS] READ CMD

Boot Loader SE Version 1.5.0 (Build ID: 1798,18531, Build Data: 2007-01-10_12:09:26)
Copyright(C) 2006 Sony Computer Entertainment Inc.All Rights Reserved.
[SERV SETCFG] XDR (CHO,CH1) ASSERT
[SERV SETCFG] XDR (CHO,CH1) DEASSERT
[INFO]: Connecting to Debug Device (SB UART)
[SERV NVS] READ CMD
[SERV NVS] READ CMD
[SERV NVS] READ CMD

```

If I do hdmi chstat 0, here is the output, still waiting for resolution:

```

Code:
>$ hdmi chstat 0
hdmi chstat 0
[HDMI] -----
[HDMI] -- HDMI Channel 0 Context --
[HDMI] -----
[HDMI]   +- [System Management]
[HDMI]     - SSM Task ID : 13
[HDMI]       * Task Status : WAITING
[HDMI]       * Wait Cause : EVENT FLAG
[HDMI]     - SSM State : WaitResolution
[HDMI]     - SSM Mutex Information : ID[2] LockTID[0] WaitTID[0]
[HDMI]     - SSM Event Flag Info : ID[2] WaitTID[13] FlagPattern[0]
[HDMI]     - SSM Mode : HDMI
[HDMI] -----
[HDMI]   - Authentication Status : NotStart
[HDMI]   - Repeater : Sink
[HDMI]   - KSVs : 0
[HDMI] -----
[HDMI]   +- [Interrupt]
[HDMI]     - External Interrupt Number of Mullion : 7
[HDMI]     - Interrupt Mask Pattern in SiI : [0x0000E0]
[HDMI]     - Interrupt Register Size : 3
[HDMI]     - Interrupt Task ID : 12
[HDMI]       * Task Status : WAITING
[HDMI]       * Wait Cause : SLEEPING
[HDMI]     - Semaphore Information : ID[38] WaitingTID[0] Count[1]
[HDMI]     - Plug Status : PowerOn
[HDMI] -----
[HDMI]   +- [SiIType]
[HDMI]     - Chip Type : 9132
[HDMI] -----
```

feng\_ye, Dec 18, 2021 [Report](#)

#2096 Like +Quote Reply



1,838

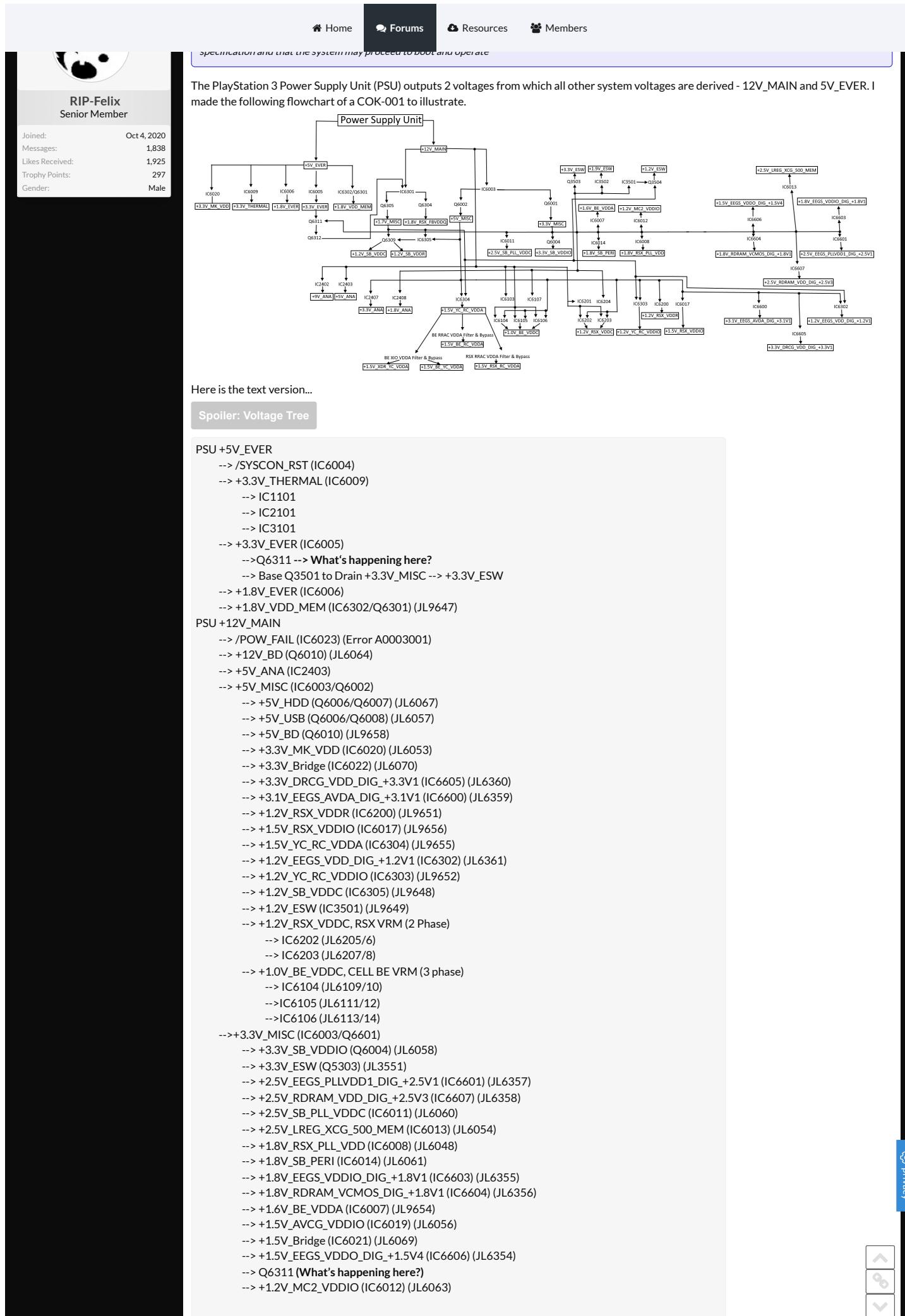


1,925

## Power Control Topology - Part 2

(Power Good & Voltage Regulation)





[Home](#)[Forums](#)[Resources](#)[Members](#)

```

--> +1.5V_YC_RC_VDDA (Q6308) (JL9655)
--> +1.2V_SB_VDDC (Q6309) (JL9648)
--> +1.2V_SB_VDDR (Q6309) (JL9648)
--> +1.2V_YC_RC_VDDIO (Q6310) (JL9652)
--> +1.2V_RSX_VDDR (Q6200) (JL9651)
--> +1.2V_EEGS_VDD_DIG +1.2V1 (Q6600) (JL6361)
--> +1.2V_ESW (Q3504) (JL9649)
--> +1.8V_RSX_FBVDDQ (IC6301/Q6304) (JL9657)
--> +1.2V_RSX_VDDC
--> IC6201 (2 Phase Buck Controller)
--> IC6202 (JL6205/6)
--> IC6203 (JL6207/8)
--> +1.0V_BE_VDDC
--> IC6103 (3 Phase Buck Controller)
--> IC6104 (JL6109/10)
--> IC6105 (JL6111/12)
--> IC6106 (JL6113/14)

```

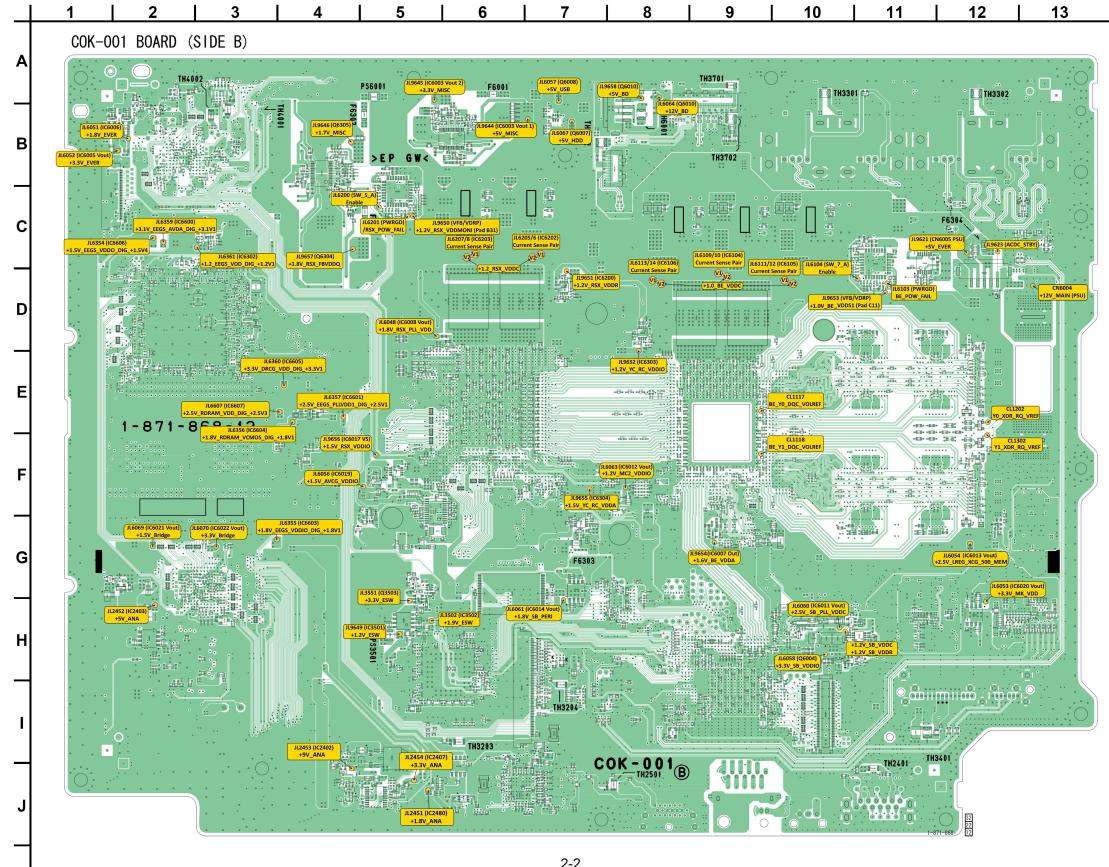
12V\_MAIN is the one from which most other system voltages are derived (5V, 3.3v, 1.8v, 1.7, 1.2, 1.0 etc). The CPU/GPU/SB, are all powered from this 12V\_MAIN. But each one of them has a chain of chips before the correct voltage can get to them. The purpose of these chips is to provide stable voltage and control of when that voltage is delivered.

For example, the CPU's 1.0V\_BE\_VDDC is produced by its Voltage Regulation Module (VRM). It includes IC6103, a buck controller which drives IC6104, IC6105, IC6106 buck converters. That controller is enabled by the syscon chip at the appropriate time to turn on the CPU. IC6107 is involved in the coordination of this effort. So too are the NEC/TOKIN bulk filtering capacitors. And also the array of MLCC bypass caps. Collectively they constitute the CPU's Voltage Regulation Module.

The RSX has it's own VRM, the SB too. Most subsystems have have an IC controller that enables their voltage at the appropriate time in the Power On Sequence. The Voltage flowchart I made above is the general manner in which these subsystems receive their voltages. Not every IC in the PS3 is listed, just the major ones involved in producing the system voltages marked by a square box. Also that image was based off the COK-001 Service Manual. It does not apply to all models, but can be used as a general guide for them. Obviously, the PS2 Hardware section will not be included in non-backward compatible models.

How does this relate to power good? Well, each system voltage in a box in that image has a controller IC that is monitoring it's output voltage. If that voltage is within regulation, it will report power good to the SYSCON "system control." If any one of them falls out of regulation and reports no power good, the syscon will refuse to boot.

The voltage flowchart above combined with the MB Side B "Jumper Lead" Test Pad locations below should make diagnosing and troubleshooting easier.



[Home](#)[Forums](#)[Resources](#)[Members](#)**Spoiler: Abbreviations**

ATA = Hard Drive Interface  
BC = PS2 Bridge Chip  
BEI = Redwood Broadband Engine Interface Controller (AKA RC)  
CE = Control Enable  
CELL BE = CPU ("Cell Broadband Engine")  
CG = Clock Generator  
CLK = Clock  
CONT = Control  
CTL = Control  
EE = Emotion Engine (PS2 CPU)  
EIB = Element Interconnect Bus  
EN = Enable  
ESW = Ethernet Switch or Switchable  
FB = Feedback  
FlexIO = Redwood Rambus FlexIO CPU/GPU Interface  
GS = Graphics Synthesizer (PS2 GPU)  
MC2 = ? (VDDIO à BE\_SPI, CHKSTP, JTAG, TBEN, & P\_L\_BYPASS)  
MIC = Yellowstone Memory Interface Controller (AKA YC)  
PCI = PS2 Hardware Interface  
PLL = Phase-Locked Loop  
PPE = Power Processor Element (main dual threaded CPU)  
PWRGD = Power Good  
RC = Redwood Broadband Engine Interface Controller (AKA BEI)  
RRAC = Redwood Rambus FlexIO CPU/GPU Interface Voltage (VDDR)  
RST = Reset  
RSX = GPU ("Reality Synthesizer")  
SB = South Bridge  
SPE = Synergistic Processing Element (8x, 1 disabled for redundancy)  
STBY = Standby  
SW = Switch  
VDD = Positive Field Emitting Transistor (FET) Voltage  
VDDA = Positive FET Voltage Supply for Analog Subsystems  
X = Rambus XDRAM Memory Subsystem  
XDR = Yellowstone XDRAMTM System Memory (Y0\_XDR0, Y0\_XDR1, Y1\_XDR0, & Y1\_XDR1)  
XGC = XDRAM Clock Generator  
XIO = Yellowstone Rambus CPU/XDR Memory Interface ("Extreme Data Rate IO")  
YC = Yellowstone Memory Interface Controller (AKA MIC)  
YRAC = Yellowstone Rambus CPU/XDR Memory Interface Voltage

**Spoiler: System Voltage Definitions**

+1.0V\_BE\_VDDC = Cell Be Processor Core (PPE/SPEs)  
+1.5V\_BE\_THERMAL\_VDDA = CPU Thermal Power?  
+1.6V\_BE\_VDDA = CPU ADC Voltage for PLL & Thermal  
+1.5V\_BE\_YC\_VDDA = CPU/XDR Yellowstone XIO Controller ADC Interface  
+1.5V\_BE\_RC\_VDDA = SB/CPU Redwood Rambus FlexIO Controller ADC Interface  
  
+1.2V\_RSX\_VDDR = RSX Redwood Rambus FlexIO Core  
+1.2V\_RSX\_VDDC = RSX Processor Core  
+1.5V\_RSX\_RC\_VDDA = RSX Redwood Rambus FlexIO Controller ADC Interface  
+1.5V\_RSX\_VDDIO = VDDP\_VO (Voltage for Picture, Video Out to DVE/HDMI)  
+1.8V\_RSX\_PLL\_VDD = RSX Phase-Locked Loop  
+1.8V\_RSX\_FBVDDQ = RSX DRAM  
  
+1.2V\_YC\_RC\_VDDIO = XDR/CPU/SB Yellowstone XIO & Redwood Rambus FlexIO Core  
+1.2V\_MC2\_VDDIO = BE\_SPI, CHKSTP, JTAG, TBEN, & P\_L\_BYPASS  
+1.5V\_XDR\_YC\_VDDA = Yellowstone Memory Interface Controller  
+1.5V\_YC\_RC\_VDDA = CPU/SB Yellowstone XIO & Rambus FlexIO Controller ADC Interface  
+1.8V\_VDD\_MEM = XDRAM Voltage  
  
+1.2V\_ESW = Ethernet Controller (IC3503) VDD\_Core (IC3501/Q3504)  
+1.9V\_ESW = Ethernet Controller (IC3503) VDDAH (IC3502)  
+3.3V\_ESW = Ethernet Controller (IC3503) VDDO (Q3501)  
  
+1.2V\_SB\_VDDC = Southbridge Processor Core  
+1.2V\_SB\_VDDR = SB Redwood Rambus FlexIO Core  
+1.8V\_SB\_PERI  
= VCC18 Starship2 Flash Controller  
= VDDP Starship2 Flash Controller  
= SB\_ATA1 (P34)  
= Power (P60)  
+2.5V\_SB\_PLL\_VDDC =Southbridge Phase-Locked Loop  
+3.3V\_SB\_VDDIO  
= VDD33 Starship2 Flash Controller  
= Nand Flash 0 & 1  
= SB\_Main EBUS (P30)  
= SB\_Peripheral Parts (P31)  
= SB\_Rear USB, ATA0, & PCI (P32)

privacy

[Home](#)[Forums](#)[Resources](#)[Members](#)

= Power (P60)

**Spoiler: SYSCON Switches (Enable & Control)**

/SYSCON\_RST --> EN Pin 3 (IC6009) --> +3.3V\_THERMAL  
 --> Vdd Pin 1 (IC1101)  
 --> Vdd Pin 1 (IC2101)  
 --> VDD Pin 8 (IC3101)

/BE\_POW\_FAIL <-- PWRGD Pin 7 (IC6103) (JL6103)

/RSX\_POW\_FAIL <-- PWRGD Pin 7 (IC6201) (JL6201)

/POW\_FAIL (12V) --> OUT Pin 4 (IC6023)

SW\_ATA --> Enables Gate Pin 2 (Q6006) --> Enables Gate (Q6007) --> +5V\_HDD

**SW\_PCI**

--> CONT Pin 5 (IC6021) --> +1.5V\_Bridge (IC7301)  
 --> CONT Pin 5 (IC6022) --> +3.3V\_Bridge (IC7301)

**SW\_0**

--> CTL1 Pin 9 (IC6003) --> +5V\_MISC  
 --> CTL2 Pin 10 (IC6003) --> +3.3V\_MISC  
 --> STBY2 Pin 10 (IC6301) --> +1.7V\_MISC

SW\_1\_A --> CONT Pin 5 (IC6020) --> +3.3V\_MK\_VDD (IC5001)

- For Clock Synthesizer

SW\_1\_B --> CONT Pin 5 (IC6013) --> +2.5V\_LREG\_XCG\_500\_MEM

- Analog Voltage for the core PLL of IC5004, which is an ICS9214 Clock Generator used to support the Rambus XDR memory subsystem and Redwood logic interface.

**SW\_2**

--> IN\_PSV Pin 1 (IC6302) --> +1.8V\_VDD\_MEM (JL9647)

**SW\_3**

--> EN Pin 3 (IC6305)  
 --> +1.2V\_SB\_VDDC (JL9648)  
 --> +1.2V\_SB\_VDDR (JL9648)

**SW\_4\_A**

--> Base Q3501  
 --> Enable Pin 3 (IC3501) --> +1.2V\_ESW  
 --> CONT Pin 5 (IC3502) --> +1.9V\_ESW  
 --> Base (Q3502) --> Base (Q3503) --> +3.3V\_ESW  
 --> P3\_ENABLE PD Pin 93 (IC3503) --> Ethernet Controller

**SW\_4\_B**

--> Enables Gate Pin 5 (Q6006) --> Enables Gate (Q6008) --> +5V\_USB  
 --> CONT Pin 5 (IC6014) --> +1.8V\_SB\_PERI  
 --> EN Pin 3 (IC6011) --> +2.5V\_SB\_PLL\_VDDC

**SW\_5\_A**

--> Enable Pin 29 (IC6201)  
 --> PWM Pin 3 (IC6202) --> +1.2V\_RSX\_VDDC (JL6205/6)  
 --> PWM Pin 3 (IC6203) --> +1.2V\_RSX\_VDDC (JL6207/8)

**SW\_6**

--> CE Pin 3 (IC6012) --> +1.2V\_MC2\_VDDIO  
 --> EN Pin 3 (IC6303) --> +1.2V\_YC\_RC\_VDDIO (JL9652)

**SW\_7\_A**

--> Enable Pin 29 (IC6103)  
 --> PWM Pin 3 (IC6004) --> +1.0V\_BE\_VDDC  
 --> PWM Pin 3 (IC6005) --> +1.0V\_BE\_VDDC  
 --> PWM Pin 3 (IC6006) --> +1.0V\_BE\_VDDC

**SW\_8\_A**

--> EN Pin 3 (IC6007) --> +1.6V\_BE\_VDDA (JL9654)  
 --> EN Pin 3 (IC6304) --> +1.5V\_YC\_RC\_VDDA (JL9655)

**SW\_8\_B**

--> EN Pin 3 (IC6019) --> +1.5V\_AVCG\_VDDIO (IC2102)  
 --> EN Pin 3 (IC6017) --> +1.5V\_RSX\_VDDIO

**SW\_8\_C**

--> EN Pin 3 (IC6008) --> +1.8V\_RSX\_PLL\_VDD  
 --> STBY1 Pin 9 (IC6301) --> +1.8V\_RSX\_FBVDDQ

privacy



[Home](#)[Forums](#)[Resources](#)[Members](#)

SW\_1.5 --> CE Pin 5 (IC6606) --> +1.5V\_EEGS\_VDDO\_DIG\_+1.5V4 (JL6354)  
 SW\_1.8 --> EN Pin 3 (IC6604) --> +1.8V\_RDRAM\_VCMOS\_DIG\_+1.8V1 (JL6356)  
 SW\_1.81 --> CONT Pin 5 (IC6603) --> +1.8V\_EEGS\_VDDIO\_DIG\_+1.8V1 (JL6355)  
 SW\_2.5 --> EN Pin 3 (IC6601) --> +2.5V\_EEGS\_PLLVDD1\_DIG\_+2.5V1 (JL6357)  
 SW\_2.65 --> CONT Pin 5 (IC6607) --> +2.5V\_RDRAM\_VDD\_DIG\_+2.5V3 (JL6358)  
 SW\_3.1 --> CE Pin 3 (IC6600) --> +3.1V\_EEGS\_AVDA\_DIG\_+3.1V1 (JL6359)  
 SW\_3.3 --> CONT Pin 3 (IC6605) --> +3.3V\_DRCG\_VDD\_DIG\_+3.3V1 (JL6360)

AUDIO Pin 16 (IC2102)

- --> RSX\_AVCLK3
- --> A1 Pin 2 (IC2105)
  - --> DRCG\_GEN18M --> Xin Pin 4 (IC7001) --> IC7002 (RDRAM)
    - --> IC7003 (RDRAM)
    - --> IC7004 (EE+GS)

Do not take the above as gospel. If you notice anything needing updated let me know and I'll edit back as necessary.

#### Getting back to Power good:

Notice the GPU Buck Controller (IC6201). It's locate at B5. This controller is what controls power to the RSX. I have labeled the jumper locations for Enable, which is the signal SYSCON sends to power on the RSX. I also labeled Power Good (PWRGD), which is the signal the controller sends back to the SYSCON about the power regulation. If it's bad, /RSX\_POW\_FAIL goes low and the syscon throws an error. Which error? 3004 or 1002. Which one is a mystery still. I hypothesize it depends upon the Step number of the Power On Sequence. Basically, when power good went low. The hypothesis is that earlier step numbers = 3004. Later step numbers = 1002. However, the earliest reported 1002 was 06, and 3004 was 09. So that is inconsistent with my hypothesis. The 06 1002 report could be inaccurate, but reports are all I have to go off of.

The last pad I labeled on the RSX VRM Controller is the voltage feedback/drop jumper. This is what the controller is monitoring to decide if power is good or not. We should be able to probe this point to see voltage drop across RSX\_VDDC. If it drops too much it triggers under voltage lock out and the controller will send no power good (PWRGD low). The syscon will error. The CPU has the same and other controllers around the board have a similar function.

Since, preventing voltage drops on the CPU/GPU is important for system stability, the NEC/TOKIN's are a concern. If they can't sustain the voltage for long enough underload, the voltage drop will fall out of regulation. One way to prevent this is to replace failing/aged tokins with new tantalum capacitors. Everyone is already familiar with that. But there is another way. By adjusting the power good voltage dropout threshold, so that there is a wider range the voltage can fall before triggering an error.

#### Adjusting Power Good Threshold

Vid pins VIDO-5 on the buck controller form a 6-digit code corresponding to the Vout No load setpoint. Power Good Vmin and Vmax thresholds are relative to that set point. With the stock COK-00X voltage divider values (15K and 20k), Vmin = -163mV. Vmax is always +100mV. The Vout voltage cannot deviate more than that. If it does power good goes low and the SYSCON will error.

$$V_{LOWER} = \frac{V_{OUTNoLoad}}{2} \times \frac{R_1 + R_2}{R_2}$$

$$V_{UPPER} = V_{OUTNoLoad} + 100 \text{ mV}$$

In some official SONY refurbished consoles, new resistor values (27K and 10K) change Vmin = -400mV. So the Low Voltage threshold is now more than twice as low, allowing much more voltage ripple before it triggers an error. My hypothesis is SONY did this to reduce the frequency of 1001 and 1002's errors. That would explain why they did it to both buck controllers (CPU and GPU). A sort of admission of guilt that they either set it too aggressively or were compensating for bad NEC/TOKINs without replacing them.

Power good low voltage threshold is there to prevent system instability, but if SONY decided it was okay to loosen it, then perhaps we can follow suit. It could be particularly important because we're seeing a lot of unexplained 1001's recently. Currently, 1002's are assumed to be bad NEC/TOKIN's. Replacing aged bulk filter capacitors certainly works, but just because changing them fixes the error doesn't mean that's the only way to skin a cat.

I have intentionally held off recommending this mod to people, because I'm only freshly aware of it's potential. Before I conclude this is okay to do, I would like to know what the VID at idle is, so we can compare actual voltage measurements with the low voltage threshold. On a console with 1001/1002 errors, Vout should drop more than that before it experiences a YLOD. That means I need oscilloscope measurements of Vout on both the CPU/GPU and PWRGD. Then, by replicating Sony's mod, I would like to see if the error goes away AND the system is stable (requires stress testing)! If so, then we have discovered an easier method of resolving these errors.

Replacing those resistors is fine micro-soldering work that pretty much requires a microscope. And I need oscilloscope measurements of the voltage drop and PWRGD (Cell BE for 1001 and RSX for 1002). So it's not too easy or cheap.

That's more than enough to digest for now.

## To be concluded...

Last edited: Feb 6, 2023

[READ THIS](#) - Links to super useful information (Schematics, pinouts, SYSCON and Frankenstein Tutorials, Tantalizer, Statistical Analysis of what really cause the YLOD, etc.)

RIP-Felix, Dec 18, 2021 [Report](#)

#2097 [Like](#) [+ Quote](#) [Reply](#)

Workz\_777, sandugas, M4j0r and 2 others like this.



@feng\_ye your software side is fine, otherwise you won't be able to see "Connecting to Debug Device (SB UART)." This is either one missing resistance from rsx side to AV ic /Hdmi ic, either that special glod. If you want to try exchange AV ic or Hdmi ic is your choice, or focus and compare that area. not sure what to say from here, but I am sure no software errors on that unit.

