OCP SUMMIT

March 20-21
2018
San Jose, CA

OPEN. FOR BUSINESS.

RunBMC - A Modular BMC Mezzanine Card

BUV - Bring Up Vehicle For BMC Mezzanine

Eric Shobe & Jared Mednick

Hardware Engineer - Salesforce

RunBMC

A Modular BMC Mezzanine

Eric Shobe & Jared Mednick, HW at Salesforce
eshobe@salesforce.com, jmednick@salesforce.com

Speaker Intro

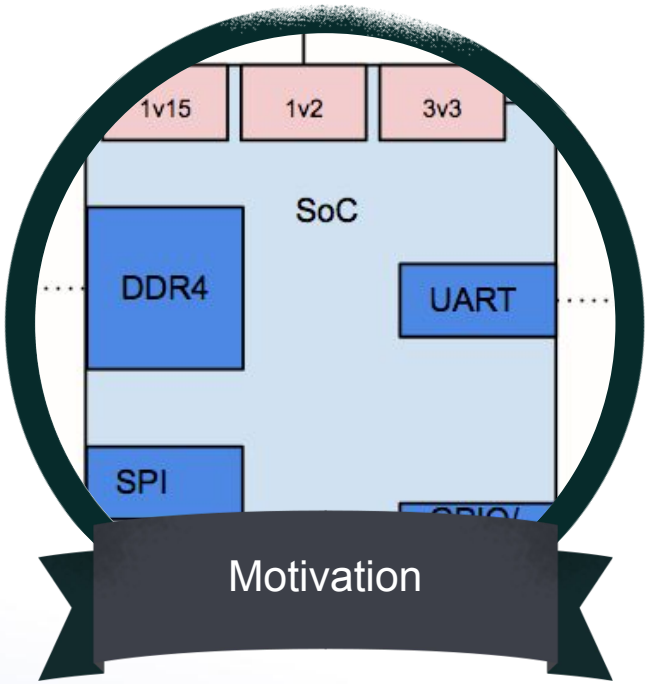
Eric Shobe

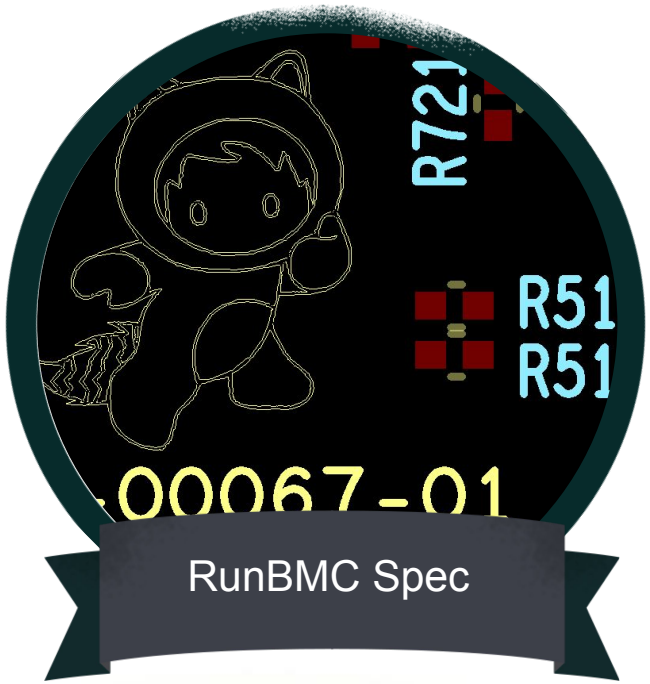
Hardware Engineer
eshobe@salesforce.com

Jared Mednick

Hardware Engineer
jmednick@salesforce.com

Overview


1v15
1v2
3v3
SoC
DDR4
UART
SPI
Motivation


R721
R51
R51
RunBMC Spec


Status


TRAIL MAP

Good News

- **Security**: OpenBMC allows users to control security issues for out of band management (faster time to patch, custom implementations of open source software, secure boot)
- **Commonality**: Hardware BMC systems generally share commonality
  - A path to a low cost open hardware platform is possible
- **Adoption:** OpenBMC management stack is expanding!

Opportunities to Improve Out of Band Management

- **Implementations**: No two out of band server management systems are exactly the same. Some implementations are proprietary, both at the chipset and the hardware level
- **Security;** Security issues require waiting on third parties to implement patches, even though many of these solutions exist today in open source
- **Manageability**: These differences require engineering efforts to adapt existing management infrastructures for new server generations, which adds delays in delivering capacity

RunBMC Benefits

Create an standard for hardware management

- Interface Consistency between vendors and implementations
- HW Security
- Supply Chain
- Can run OpenBMC management stack!
- An adaptable software platform

Baseboard Management Controller - Software

Software Stack

- Different software stacks in a BMC (Serial-Over-Lan, Sensors, Power Control, FRU's, NC-SI, LAN Software, IPMI, USB, SNMP traps, Fan Speed/Control, Event Logs, etc)

Benefits of control and standardization, some examples

- Cost Savings to control your code
- Sensor Data
- Security
- Leverage your base code
- Increase of OpenBMC contributions

Baseboard Management Controller - Hardware

OCP and ODM/OEM systems share a common subsystem for the BMC.

Typical Hardware Feature set:

- SoC
- DDR4 (256 or 512MB) , PCIe Gen2
- 2x SPI NOR Flash for BMC FW (32MB), 2x for BIOS FW
- 1x eSPI bus (Communication w/ CPLD or PCH)
- Multiple I2C bus (FRU, sensors), Multiple UARTs (OOB access), Multiple USB 2.0
- JTAG/LPC, GPIOs/TACH/PWM, VGA (sometimes)
- RGMII for 1GbT Ethernet, RMII for NC-SI or 100Mb

Baseboard Management Controller

Platform Analysis

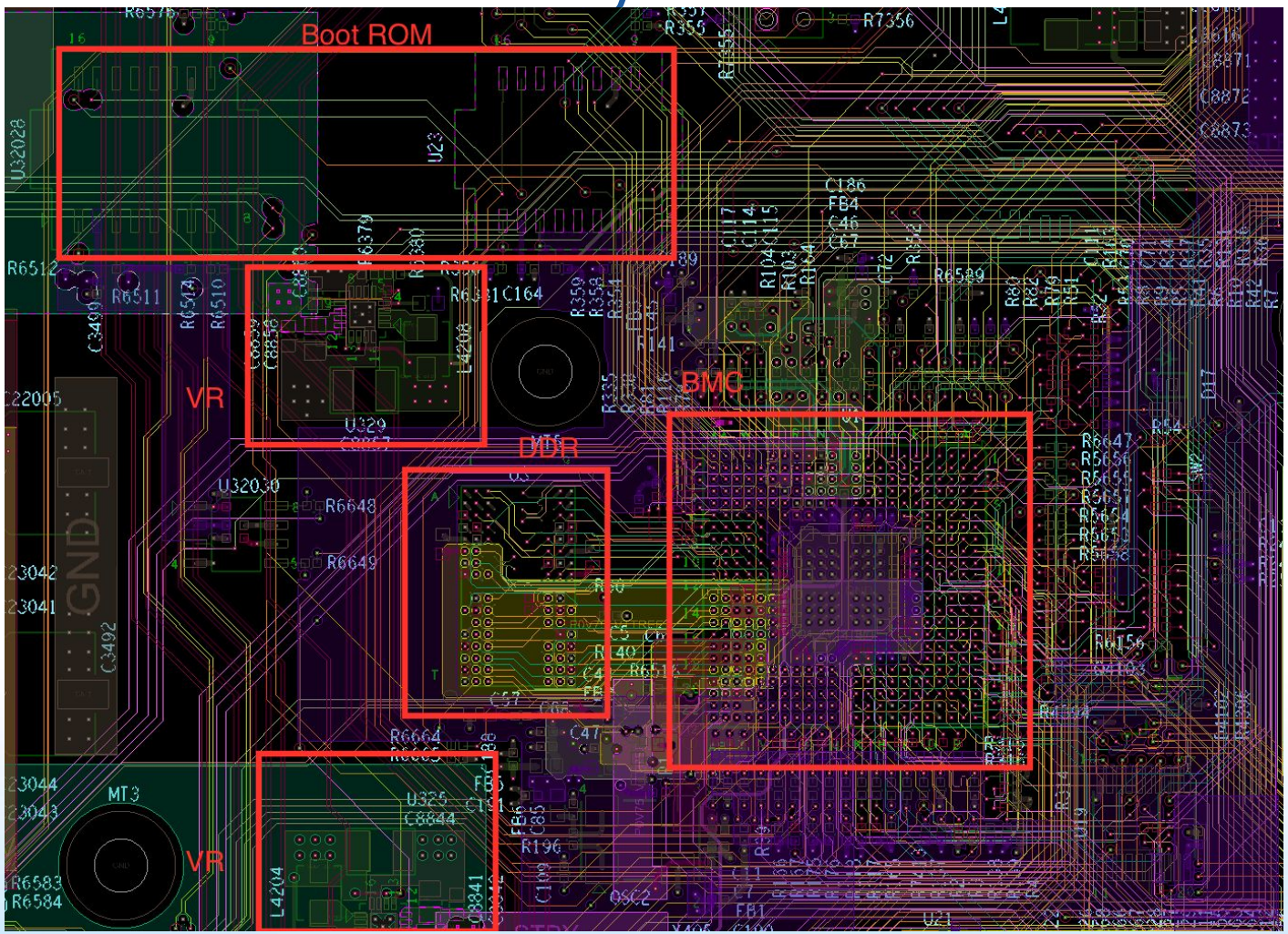
- In depth analysis motivated the modular concept
- Next slides compare the different design topologies

BaseBoard Management Controller - Platform Analysis

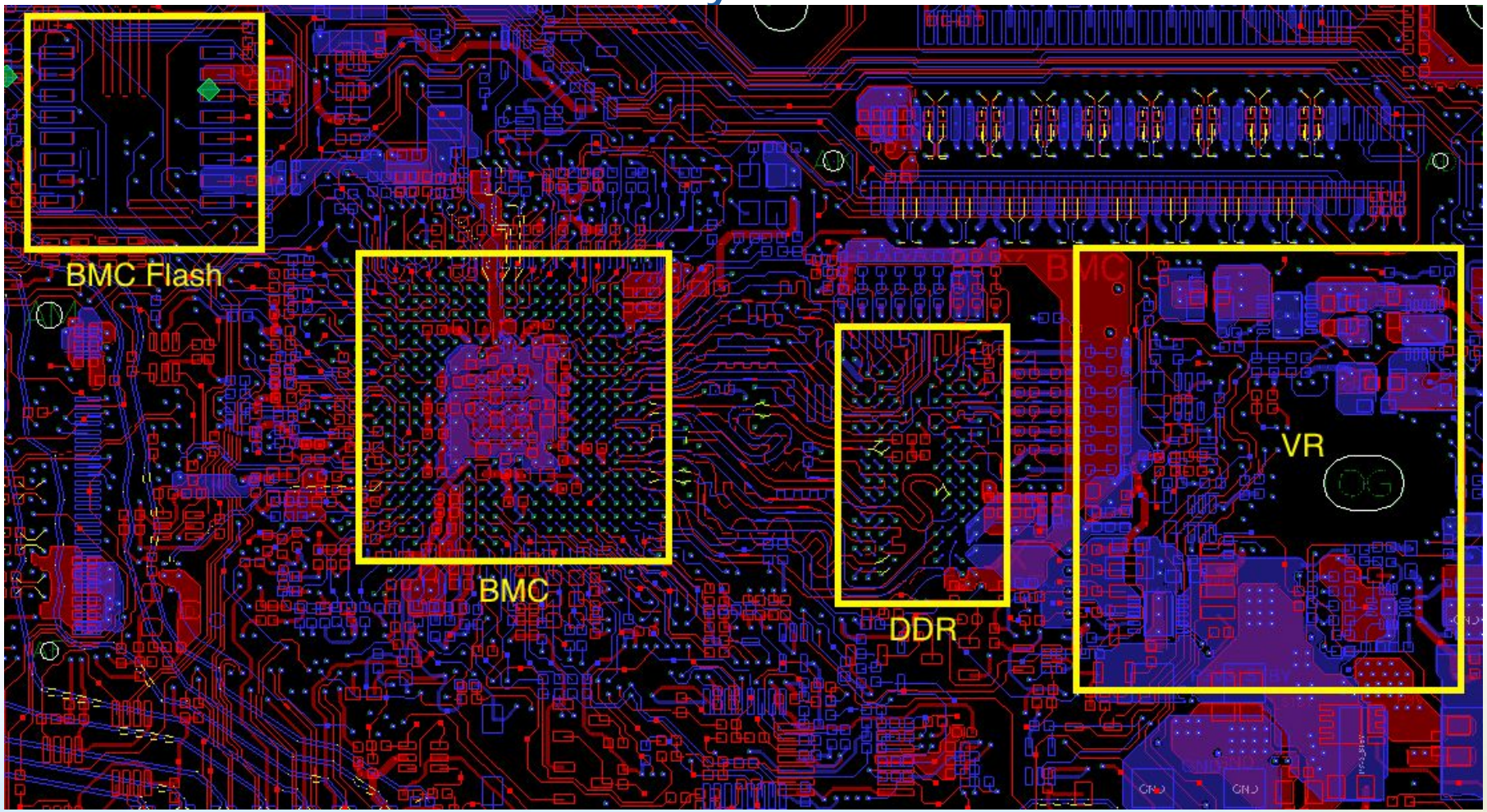
	GPIOs	IRQs	Total (GPIOs/IRQs/Others)
Server #1	58	32	103
Server #2	71	20	98
Server #3	92	8	111

Much of this signal functionality is typically shared with the PCH (for example, control of Host voltage supply)

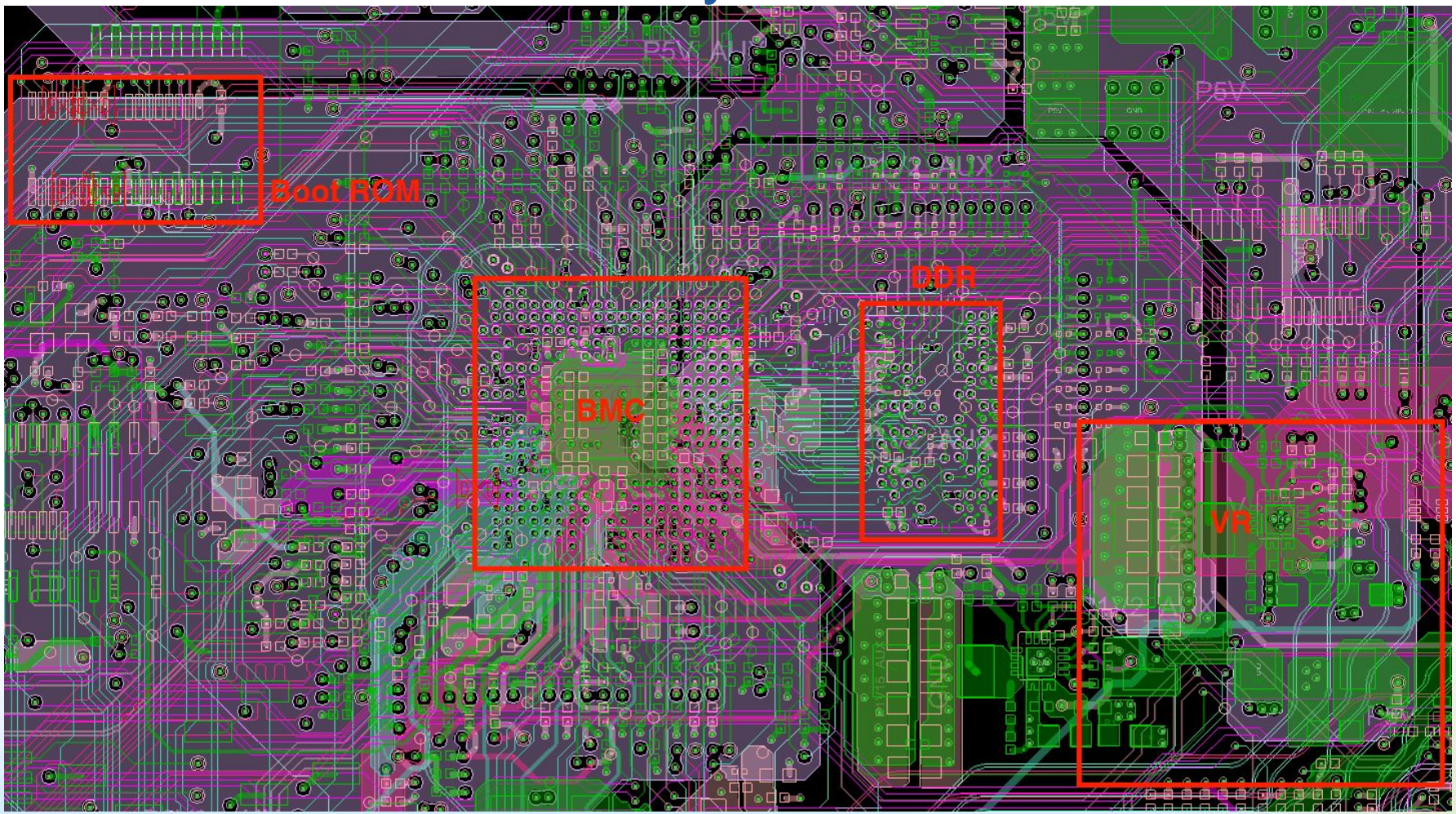
Switch BMC Subsystem



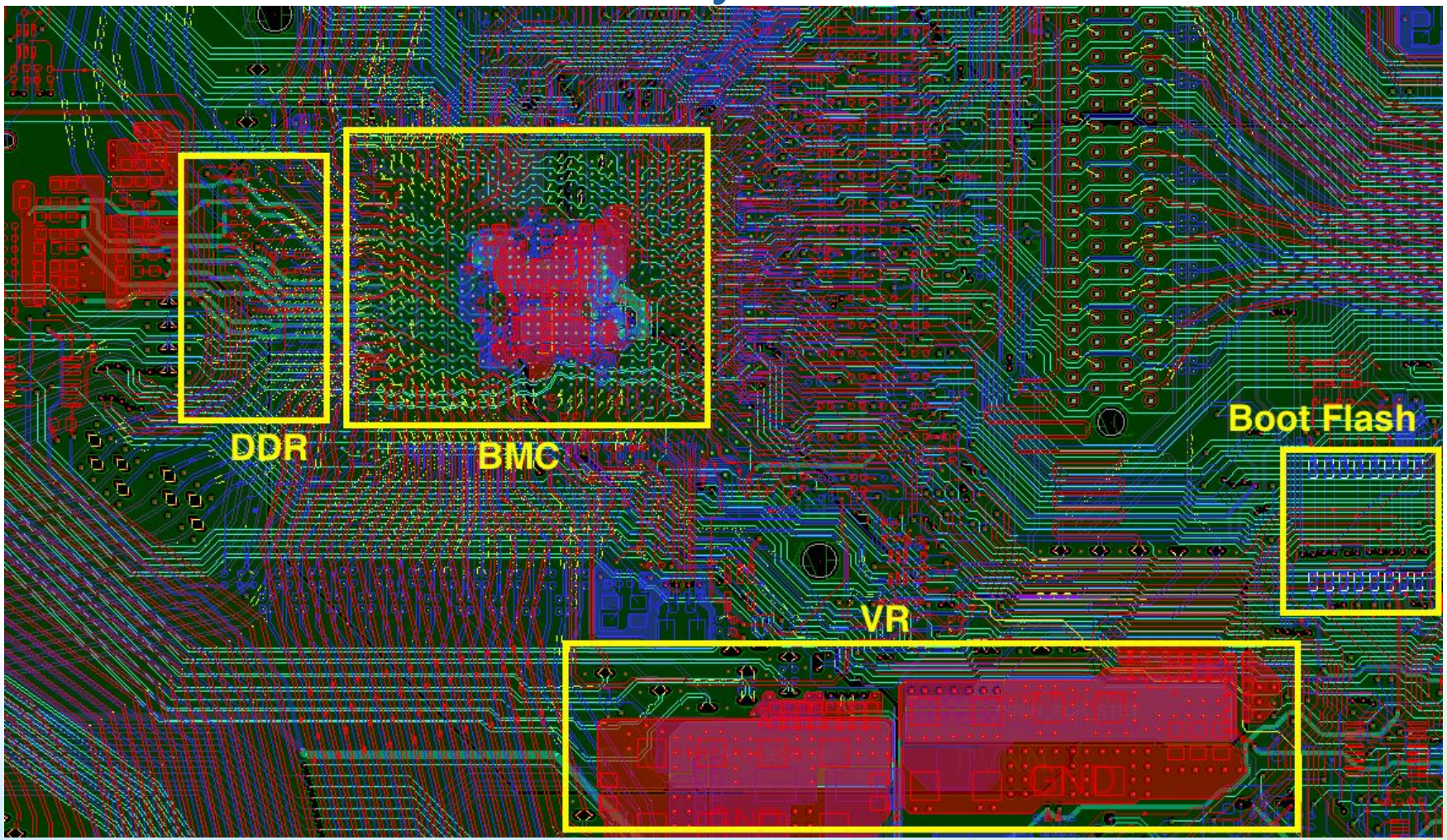
OCP Server BMC Subsystem #1



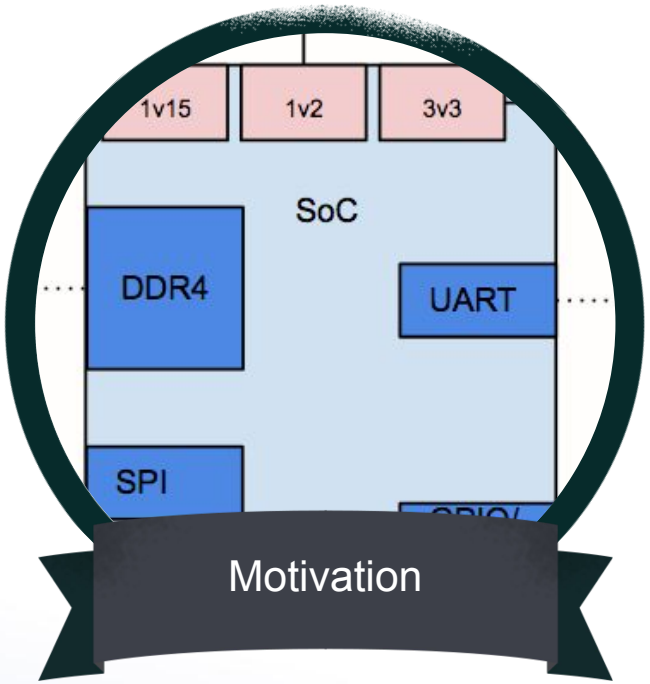
OCP Server BMC Subsystem #2

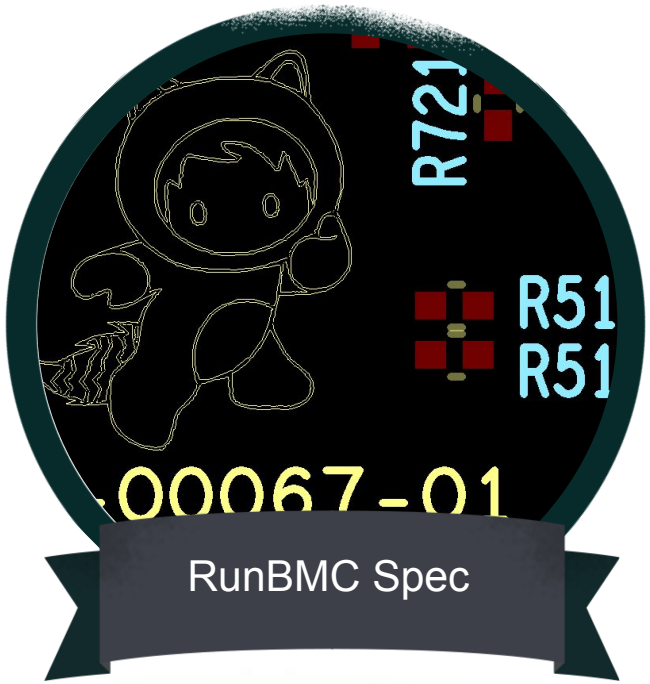


OCP Server BMC Subsystem #3



Overview


1v15
1v2
3v3
SoC
DDR4
UART
SPI
Motivation


R721
R51
R51
RunBMC Spec


Status

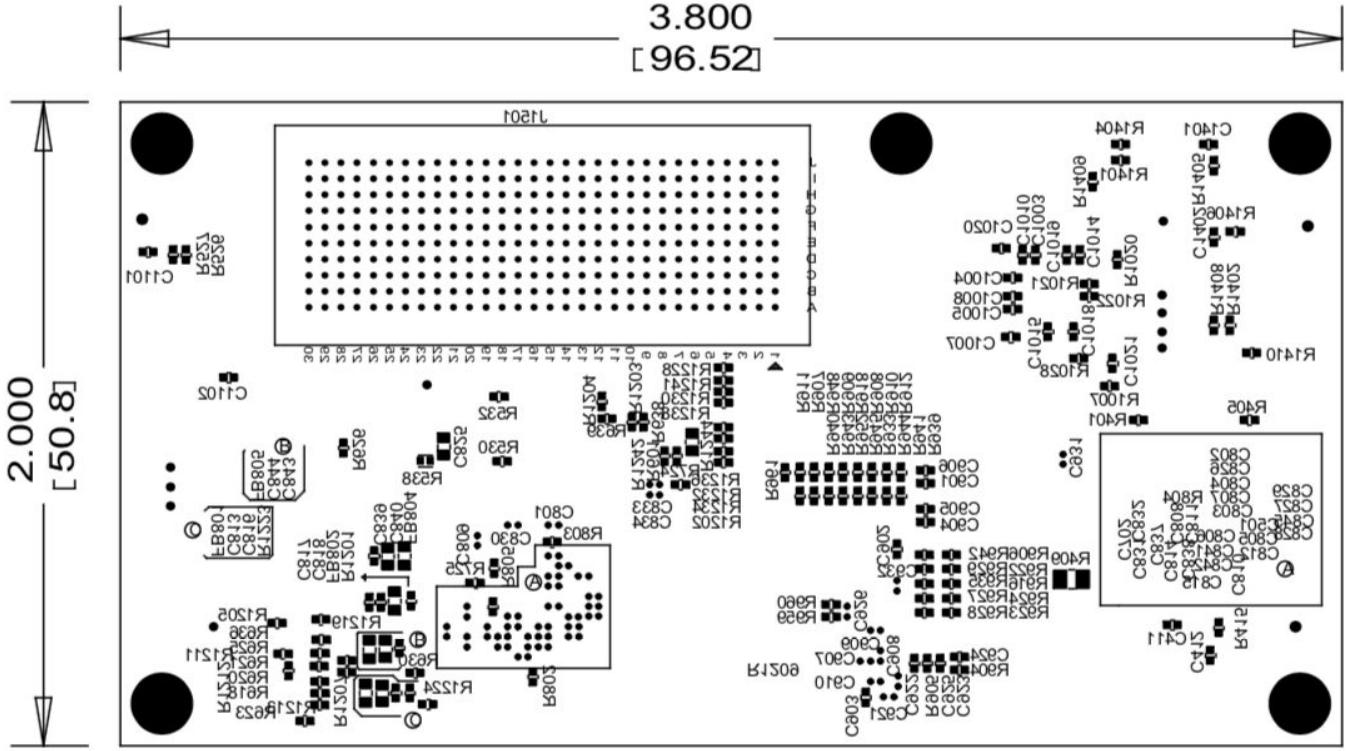
RunBMC Specification Details

Proposal

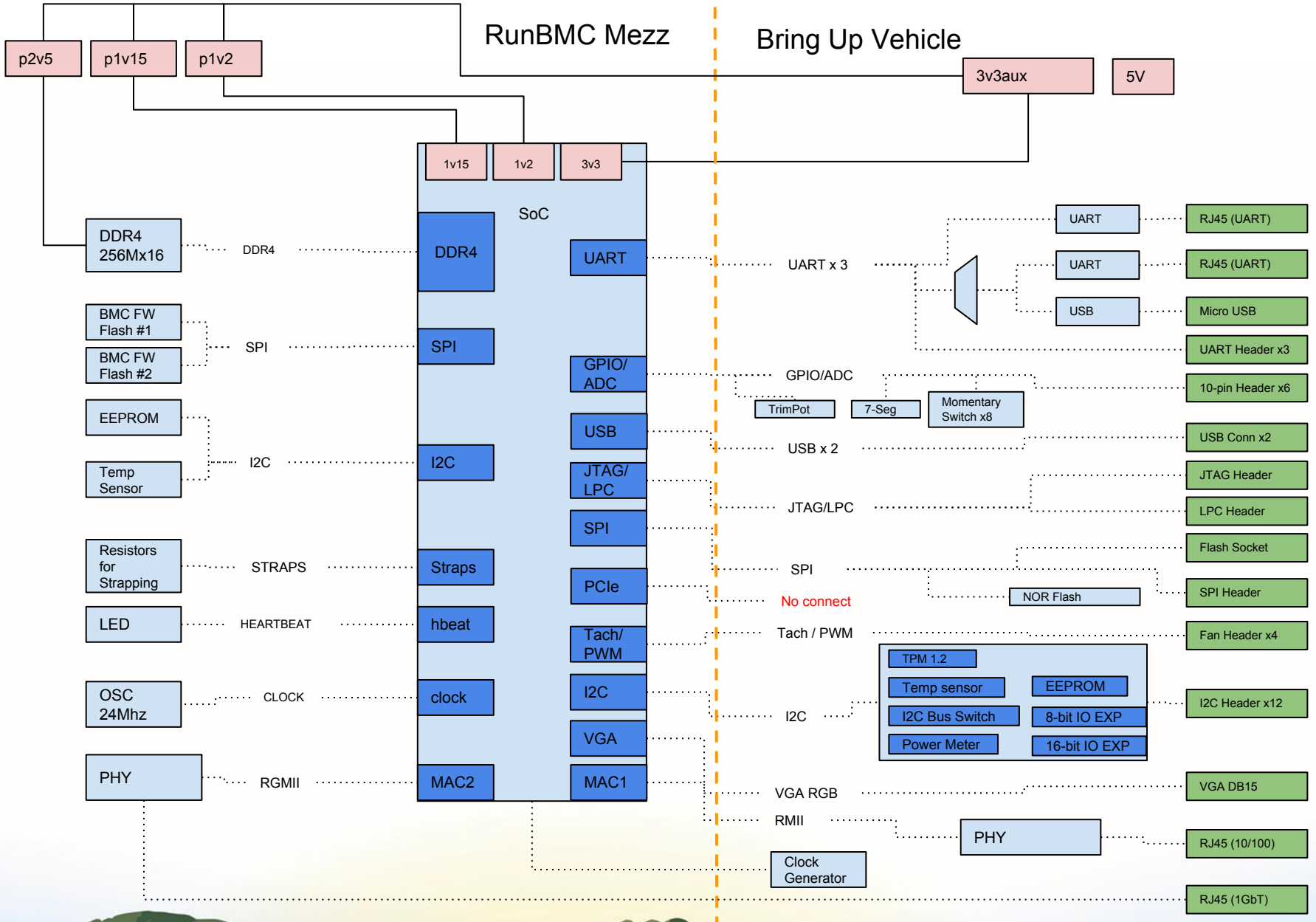
- Modular Design
- Standardize BMC to HW interface
- Standardize the HW footprint

Collaboration!

- Work together to refine spec



RunBMC


RunBMC Mezz
Bring Up Vehicle
p2v5
p1v15
p1v2
3v3aux
5V
1v15
1v2
3v3
SoC
DDR4 256Mx16
DDR4
DDR4
UART
UART x 3
UART
RJ45 (UART)
UART
RJ45 (UART)
USB
Micro USB
UART Header x3
BMC FW Flash #1
BMC FW Flash #2
SPI
SPI
GPIO/ ADC
GPIO/ADC
TrimPot
7-Seg
Momentary Switch x8
10-pin Header x6
EEPROM
Temp Sensor
I2C
I2C
USB
USB x 2
USB Conn x2
JTAG/ LPC
JTAG/LPC
JTAG Header
LPC Header
SPI
SPI
Flash Socket
NOR Flash
SPI Header
Resistors for Strapping
STRAPS
Straps
PCIe
No connect
LED
HEARTBEAT
hbeat
Tach/ PWM
Tach / PWM
Fan Header x4
TPM 1.2
Temp sensor
EEPROM
I2C Bus Switch
8-bit IO EXP
Power Meter
16-bit IO EXP
I2C Header x12
OSC 24Mhz
CLOCK
clock
I2C
I2C
VGA
VGA RGB
VGA DB15
PHY
RGMII
MAC2
MAC1
RMII
PHY
RJ45 (10/100)
Clock Generator
RJ45 (1GbT)

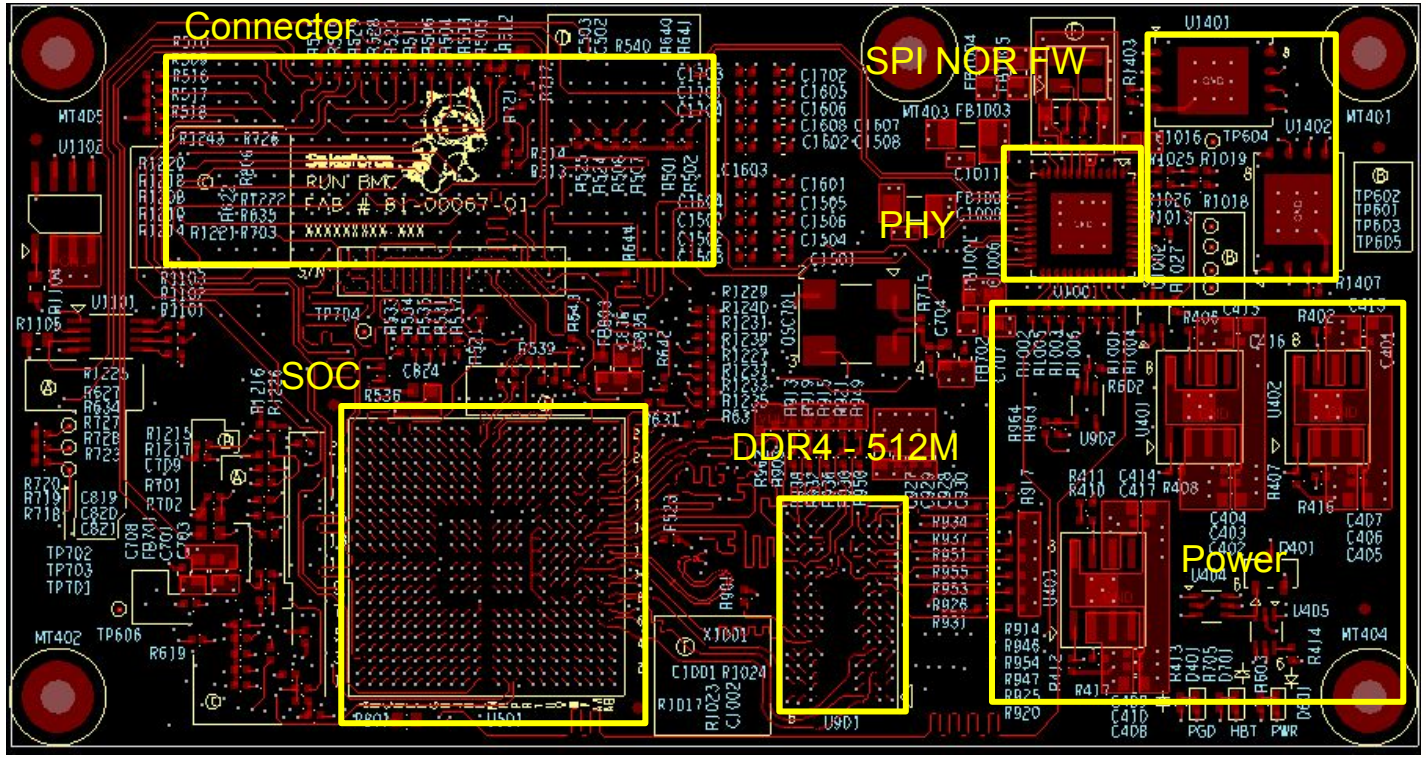
RunBMC - Connector Pinout, optimize

	A	B	C	D	E	F	G	H	I	J
1	3.3V	3.3V	3.3V	3.3V	3.3V	3.3V	3.3V	3.3V	3.3V	3.3V
2	3.3V	3.3V	3.3V	3.3V	3.3V	3.3V	3.3V	3.3V	3.3V	3.3V
3	GND	GND	GND	GND	GND	GND	GND	GND	GND	GND
4	ADC11	PHY_LED1	GND	GND	GND	GND	GND	GND	PEREFCLKP	GND
5	ADC12	GND	LPC_CLK	GPIOM7	PWM0	GPIOH5	I2C1_SCL	GND	PEREFCLKN	PHY_MDI_P0
6	ADC13	PHY_LED2	LPC_FRAME_N	GPIOF0	PWM1	GPIOH6	I2C1_SDA	UART_TXD4	GND	PHY_MDI_N0
7	GND	GPIOL2	LPC_SERIRQ_N	GPIOF1	PWM2	GPIOH7	GND	UART_RXD4	PERXP	GND
8	GPIOQ4	GPIOL3	LPC_AD0	GND	GND	GPIOJ0	I2C5_SCL	GND	PERXN	PHY_MDI_P1
9	PECIVDD	GPIOL4	LPC_AD1	GPIOF2	PWM3	GND	I2C5_SDA	UART_TXD1	GND	PHY_MDI_N1
10	GPIOQ5	GPIOE0	LPC_AD2	GPIOF3	GND	SYSCK	GND	UART_RXD1	PERST_N	GND
11	GPIOQ6	GPIOE1	LPC_AD3	GPIOF4	PWM4	SYSMOSI	I2C11_SCL	GND	PECI	PHY_MDI_P2
12	GPIOQ7	GND	GND	GPIOH0	PWM5	SYSMISO	I2C11_SDA	UART_RTS4	GPIOD6	PHY_MDI_N2
13	GND	GPIOE2	SPI2_CS1_N	GPIOH1	PWM6	SYSCS_N	GND	UART_RTS1	GND	GND
14	ADC14	TACH14	JTAG_TDI	GPIOH2	PWM7	GND	I2C2_SCL	GND	JTAG_TRST	PHY_MDI_P3
15	ADC15	TACH15	JTAG_TMS	GPIOH3	SPI2_CS0_N	GPIOJ1	I2C2_SDA	UART_TXD3	MDIO1	PHY_MDI_N3
16	GND	GPIOE3	GND	GPIOH4	GND	GPIOJ2	GND	UART_RXD3	GND	GND
17	SPI2_SCK	LPC_RST_N	JTAG_TCK	GND	RMII1TXEN	GPIOJ3	I2C3_SCL	GND	GPIOD7	MDC1
18	GPIOL0	GPIOE4	JTAG_TDO	DACB	RMII1TXD0	GPIOD0	I2C3_SDA	UART_TXD5	GPIOA0	ADC4
19	GND	GPIOE5	GND	DACG	RMII1TXD1	GPIOD1	GND	UART_RXD5	GND	GND
20	RMII1RCLKI	GND	GND	DACR	GPIOT4	GPIOD2	I2C4_SCL	GND	PETXP	ADC0
21	GPIOU5	PHY_LED3	TACH0	VGAHS	GPIOT5	WDTRST1	I2C4_SDA	I2C12_SDA	PETXN	ADC2
22	RMII1RXD0	GPIOM0	TACH1	VGAVS	GND	WDTRST2	GND	I2C12_SCL	GND	ADC1
23	RMII1RXD1	GPIOM1	TACH2	GND	JTAG_RTCK	GND	I2C7_SCL	GND	RST_N_CONN	GND
24	RMII1CRSDV	GPIOM2	TACH3	DDCCLK	GPIOD4	SPI1CK	I2C7_SDA	I2C10_SDA	ADC5	ADC8
25	RMII1RXER	GPIOM3	TACH4	DDCDAT	SPI2_MOSI	SPI1CS0_N	GND	I2C10_SCL	ADC6	GND
26	GPIOL1	GPIOM4	TACH5	GND	SPI2_MISO	SPI1MISO	I2C8_SCL	GND	ADC7	ADC9
27	GND	GPIOM5	TACH6	TACH10	GND	SPI1MOSI	I2C8_SDA	I2C6_SDA	ADC3	GND
28	USB2B_DP	GND	TACH7	TACH11	USB2A_DP	GND	GND	I2C6_SCL	GND	ADC10
29	USB2B_DN	GND	TACH8	TACH12	USB2A_DN	GND	I2C9_SCL	GND	GPIOA1	GND
30	GND	GPIOM6	TACH9	TACH13	GND	GPIOD3	I2C9_SDA	GPIOD5	GPIOA2	GPIOA3

RunBMC - Reference Board

Features

- SOC
- DDR4 (512MB)
- 300 pin 1.27mm pitch connector
- 1Gb Phy for RGMII
- Power Delivery standardized
- 2x SPI NOR for BMC FW
- Local I2C
- 3x UARTs, 2x USB, JTAG, LPC
- 2x SPI
- Tach/PWM
- 12x I2C
- VGA
- 1GbT and NC-SI



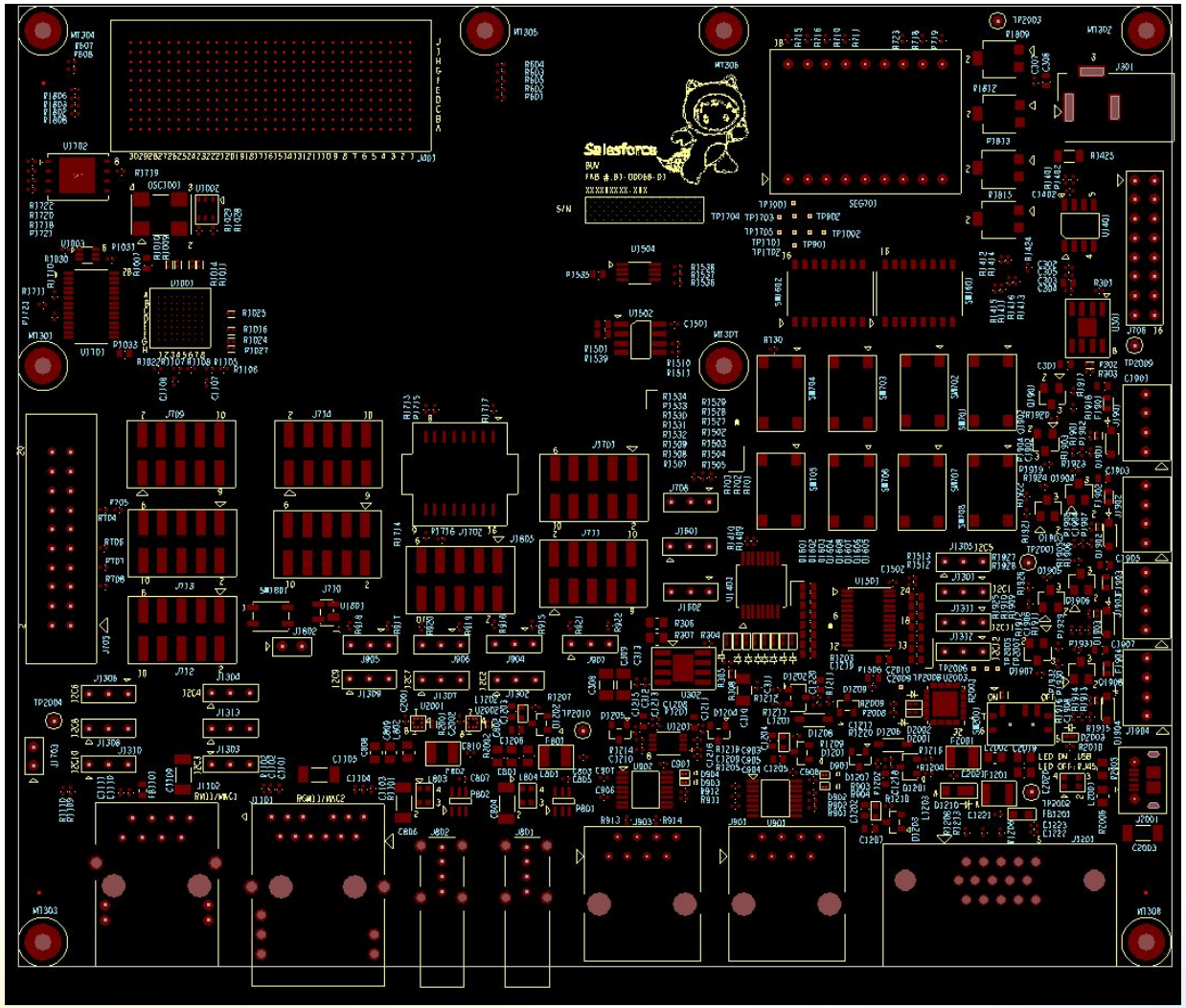
BUV - Bring Up Vehicle

Our Solution

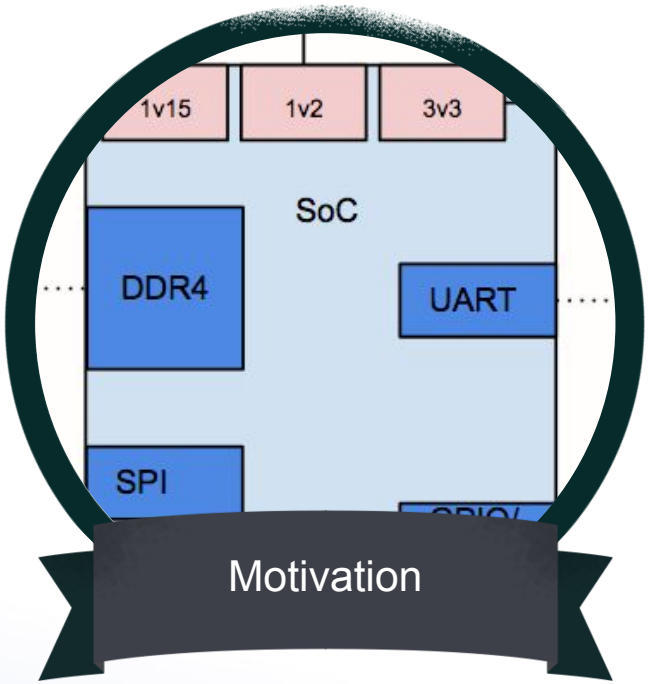
- Bring-up board allows easy development cycles w/out hardware overhead. Access to network interfaces, UART, some basic parts for testing. Can be used at your desk, universities, etc

Features

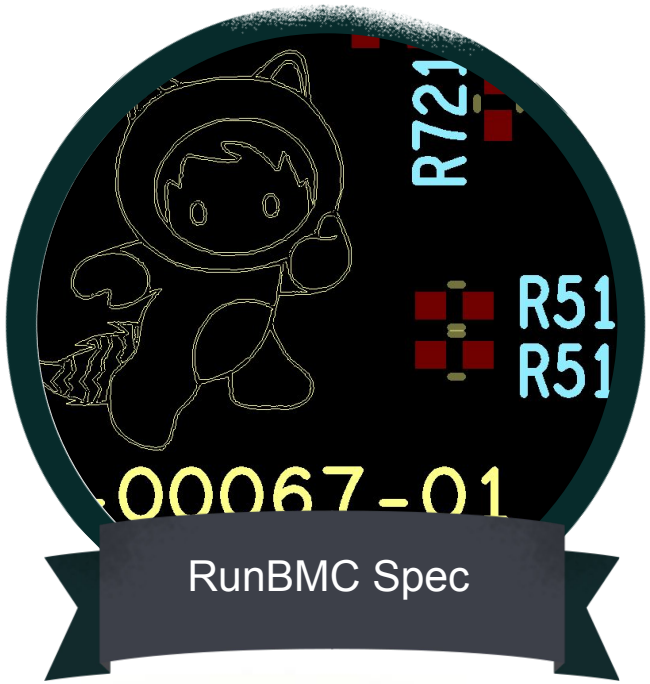
- RMII to 10/100 PHY
- 2x RJ45 for 1Gb and 100Mb
- Micro-USB access for console and power
- LPC, JTAG headers
- 12x I2C headers
- GPIO/ADC headers
- 7 seg display
- 3x console
- JTAG/LPC
- TPM
- 2x SPI bus
- TACH/PWM
- I2C devices (TPM, Temp, Current, expander)
- VGA
- Clocks



Overview



Motivation



RunBMC Spec

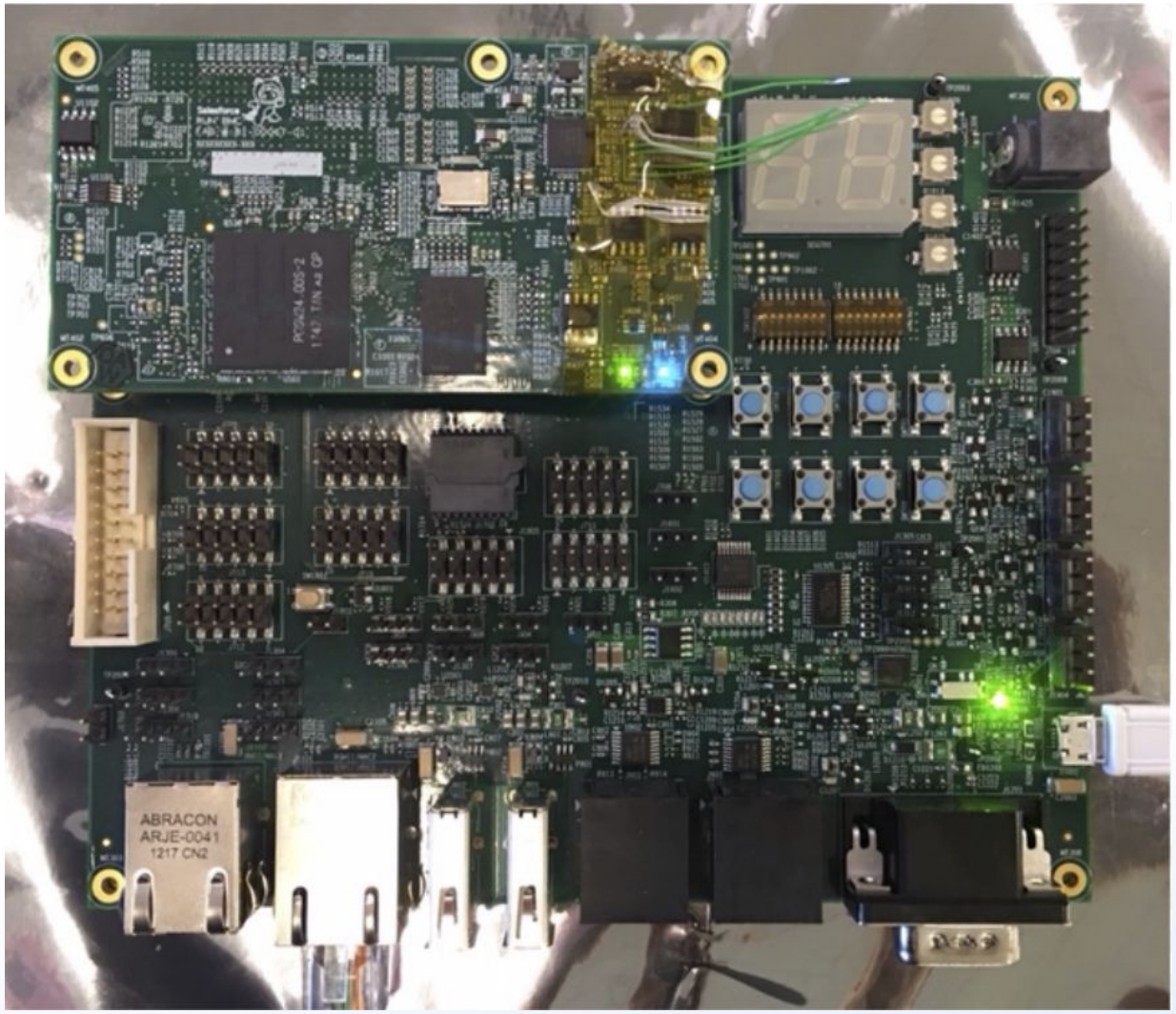
Status

RunBMC Mezzanine - Stage I | RunBMC + Bring Up Vehicle

Goal is to create a stand-alone BMC mezzanine (RunBMC) + bring-up board. Low risk project compared to full platform port (stage II). Success is defined by complete bring-up of board, booting linux, and ssh'ing into BMC.

RunBMC mezzanine board	A re-usable daughter card containing the BMC chip, common subsystems (i2c, power, ddr, clocks, eeproms, flash), and a small pitch connector for any hardware platform to use (switches, servers, etc). Only the platform SW changes.
Bring-up vehicle board	RunBMC mezzanine and Bring-up board allows easy development cycles w/out hardware overhead. Access to network interfaces, UART, some basic parts for testing. Can be used at your desk, universities, etc
Re-use	RunBMC mezzanine would allow faster turnaround from a hardware development life-cycle, earlier platform software development, and higher software adoption from developers (using the development bring-up board)

RunBMC - Bring-Up Success!




```
runBMC#
runBMC#
runBMC# bdinfo
arch_number = 0x000022B8
boot_params = 0x80000100
DRAM bank   = 0x00000000
-> start    = 0x80000000
-> size     = 0x1F000000
ip_addr     = 192.168.1.2
baudrate    = 115200 bps
TLB addr    = 0x9EFF0000
relocaddr   = 0x9EFAA000
reloc off   = 0x9EFAA000
irq_sp      = 0x9EB88EF0
sp start    = 0x9EB88EE0
runBMC#
Meta-Z for help | 115200 8N1 | NOR |
```

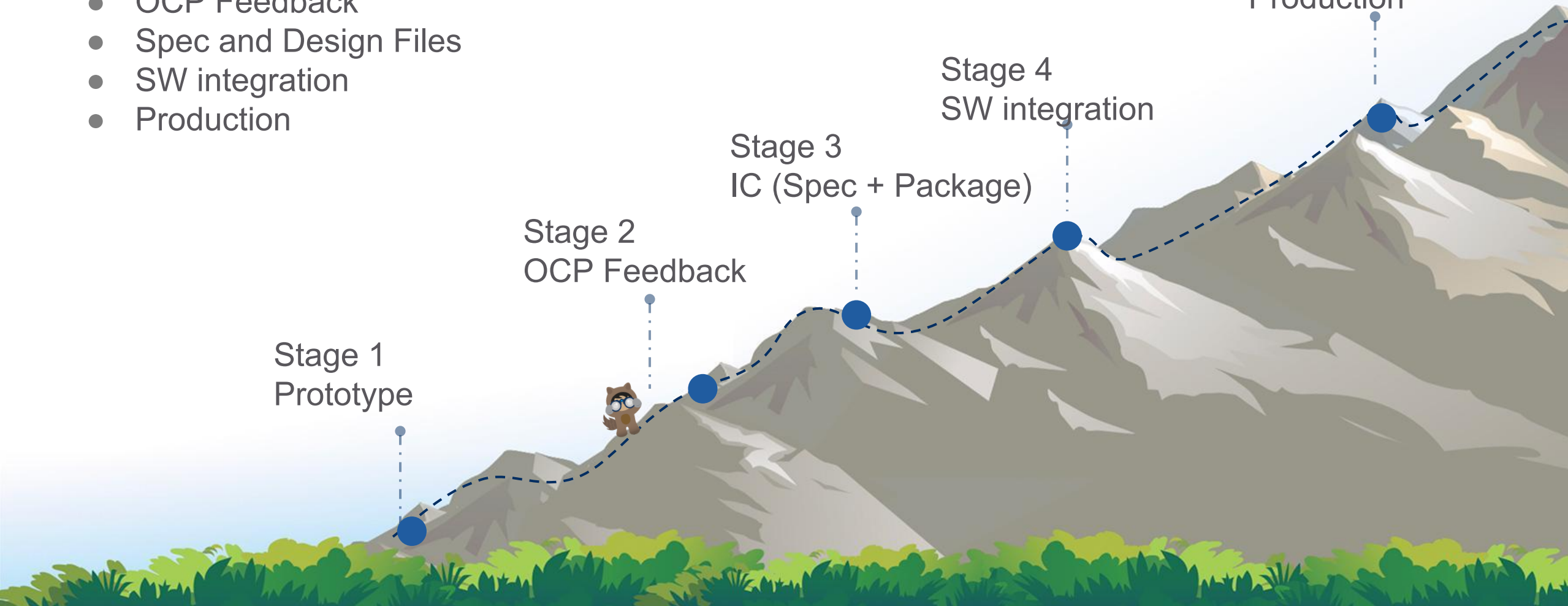
RunBMC Mezzanine - Stage II | RunBMC + Platform Port

Integrate RunBMC mezzanine with ODM server platform of choice. Riskier port due to interactions with CPU, PCH, CPLD/FGPA subsystems, power-on, and deeper software integration. However - lots of code will be re-used.

RunBMC mezzanine board	The re-usable daughter card defined in Stage I.
ODM server platform	Our ODM platform modified (schematic/layout) to use the BMC mezzanine board.
Software Integration	Deeper integration to SFDC tools (agents on the BMC)

Milestones

- Prototype + Design packages
- OCP Feedback
- Spec and Design Files
- SW integration
- Production



Citations

- Server Layout Images:
  - http://opencompute.org/wiki/Server/SpecsAndDesigns

Questions?


THANK YOU

OCP SUMMIT