

## 1013 D000 2.5K 1013 C60 Core Global Timer (1K) 1013 C200 1013 C10 Core SCU (256B) 1013 COO L2C Controller (16K) Reserved (15M) 1013 800 1210 000 DDR\_PUBL (16K) E000 0000 SMC Bank1 (1M) Reserve (32K) 2004 000 1200\_000 1013 000 APB GPIO2 (8K) CPU AXI BUS (32K) 2003\_E00 1110\_000 000F\_FFFF/110F\_FFF 1012 800 APB GPIO1 (8K) 6000 0000 SMC Bank0 (1M) APB UART1 (8K) 0000\_0000/1100\_000 2003\_C00 1012 600 APB Timer1 (8K) APB UARTO (8K) Reserved 1012 400 1050 400 200A 000 APB Timer0 (8K) Reserved (16K) NandC (16K) 2003\_800 1012\_0000 1050\_0000 APB GPIO0 (16K) Reserved (1M) 2003 400 1011 E00 I2S2/PCM(2ch) (8K) Peri AXI BUS (1M) APB PWM0/1 (16K) 2003 000 I2S1/PCM(2ch) (8K) APB I2C1 (8K) 2002 E00 1022 C000 12S0/PCM(8ch) (8K) AHB Arbiter1 APB 12C0 (8K) APB SMC Reg (16K) 1022\_8000 2002 C00 2007 C000 HDMI TX (8K) AHB Arbiter0 (16K) APB DMAC2 Reg (16K) 2002 400 2007\_800 RGA (8K) PID\_FILTER (16K) DDR\_PCTL (16K) 1022 0000 2002\_000 1011\_4000 IPP (16K) APB DMAC1 Reg (16K) eMMC (16K) 2001 COO B Secur LCDC1 (8K) SDIO (16K) APB SAR-ADC 1010\_E000 2001\_800 SD/MMC0 (16K) APB TZPC (16K) APB UART3 (16K) 1010 C000 2001 400 2006\_8000 CIF1 (8K) HSADC/TSI (16K) APB eFuse (16K) 1010 A000 2001\_000 2006 4000 GPS (16K) APB Timer3 (8K) APB TSADC (16K) 1010\_8000 2000 E00 2006\_0000 VCodeo (16K) Reserved (16K) APB I2C4 (16K) 1010 4000 1020\_8000 2000 C00 APB GPIO6 (8K) MAC (16K) APB 12C3 (16K) 1010\_0000 2000 A00 Reserved (16K) APB GRF (8K) Reserved (384K) 100a 0000 1020\_0000 2000 800 2005 4000 APB PWM2/3 (16K) USB HOST2.0 (256K) APB PMU (16K) 1009\_000 101C 0000 2000\_400 2005 0000 APB WDT (16K) USB OTG (256K) APB CRU (16K) 1008 000 1018 0000 2004 C000 CPU Debug (128K) Reserved (32K)

## BTMODE = High Level

Fig. 2-3 RK30xx Address Mapping when BTMODE=high

1013 E000

1FFE\_000

## 2.2 System Boot

1000 0000

RK30xx provides system boot from off-chip devices such as 8bits/16bits async nand flash, spi and emmc memory. When boot code is not ready in these devices, also provide system code download into them by usb otg and uart interface. All of the boot code will be stored in internal boot rom or external 8bits nor flash device, which is decided by input level of external input pin BTMODE. The following is the whole boot procedure for boot code, which will be stored in bootrom or nor flash in advance.

2004\_4000

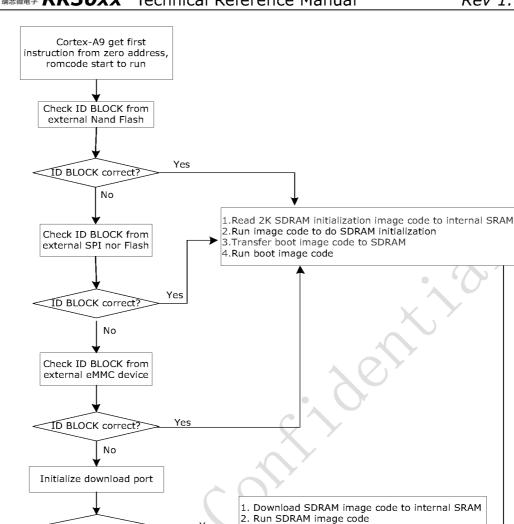


Fig. 2-4 RK30xx boot procedure flow

Yes

3. Wait request for download loader image code4. Download loader image code to SDRAM

os,

Boot or download end

5. Run loader image

## 2.3 System Interrupt connection

Any request on the UART interface?

Initialize USB port

1.Wait request for download SDRAM image code 2.Download SDRAM image code to internal SRAM

4. Wait request for download loader image code

5.Download loader image code to SDRAM

3.Run SDRAM image code

6.Run loader image

No

RK30xx provides an general interrupt controller(GIC) for Cortex-A9 MPCore processor, which has 76 SPI interrupt sources and 3 PPI interrupt source and separately generates one nIRQ and one nFIQ to CPU. The triggered type for each interrupts is high level sensitive, not programmable. The detailed interrupt sources connection is in the following table. For detailed GIC setting, please refer to Chapter 12 .