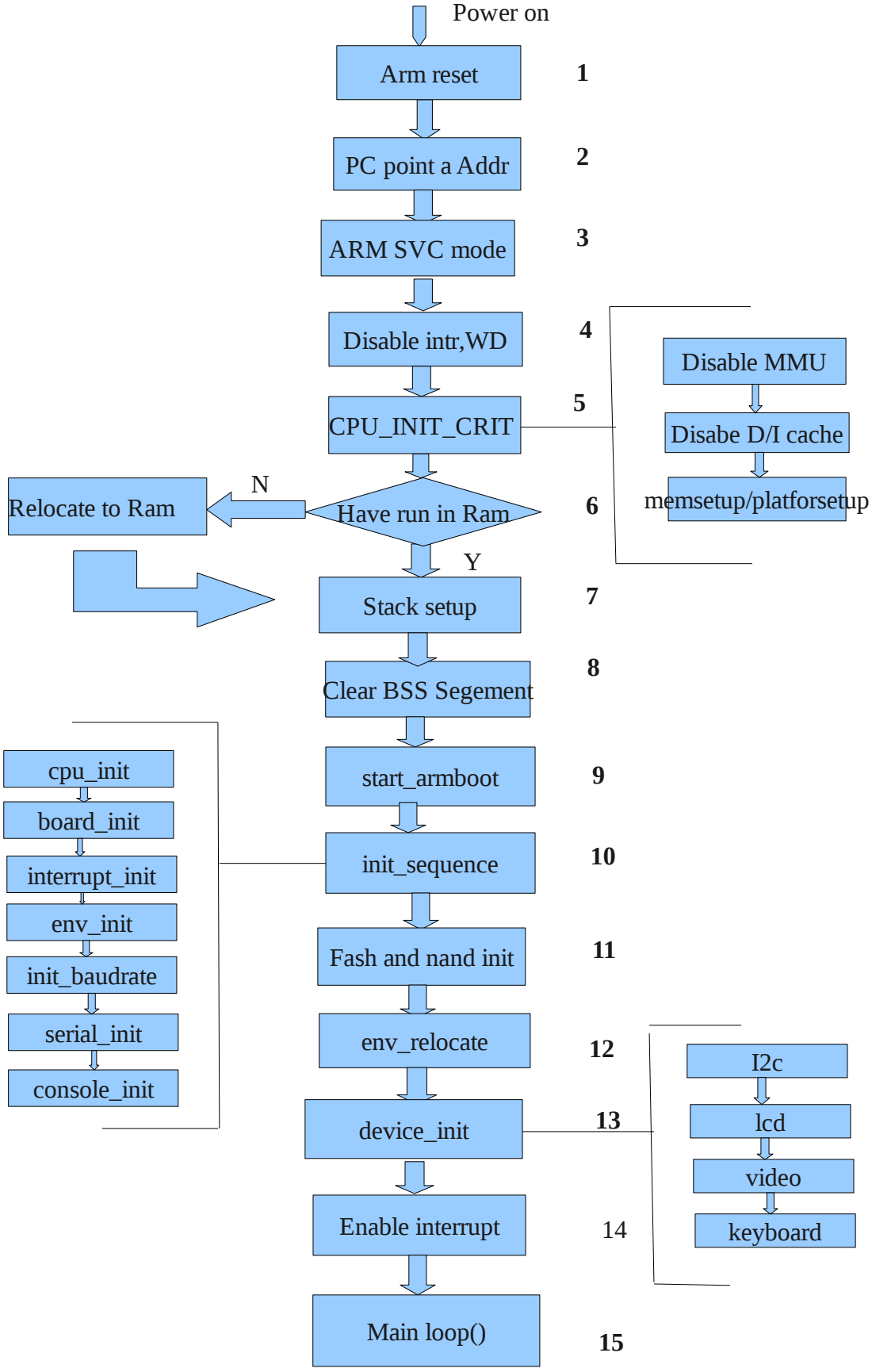
ARM u-boot initialization flow

I will use a schematic to describe how the ARM arch is booted by u-boot(no RBL)



Step one:

The ARM will fetch the first instruction from the appoint address,so the u-boot executable file will be located the address.

Step two &Three:

The first instruction is a jump action ,The ARM CPU will turn into SVC mode.

Step Four:

The u-boot will disable all interrupts and watchdog .

Step Five:

The u-boot will disable MMU unit ,D/I cache,and then invoke MEMSETUP function to configure RAM. Of course,you can initiate other memory chip set ,such as SMC,DDR and so on.

Step Six:

The U-boot program will copy itself from flash memory to a appoint address in RAM. If the u-boot executable code was saved in the NAND Flash memory,we must initiate NAND FLASH before copying action. Of course if you the U-boot have been in ram, we wouldn't do this action.

Step Seven:

The SP pointer will be assigned in this step. Which is preparing for calling “C” codes.

Step Eight:

clear BSS segment.

Step Nine:

The PC pointer will be assigned the address value of “start_armboot” in “C” codes.

Step Ten:

The INIT_SEQUENCE function will do cpu_init,board_init,interrupt_init,env_init,init_baudrate,serial_init and display some system configuration inforamations.

cpu_init: This function would allocate IRQ &FIQ stack memory if you define.

board_init: This function will initiate GPIO and turn on led

interrupt_init: This function will initiate timer controller.

evn_init: This function will initiate the address of env

init_baudrate: Just initiate the baud rate.

serial_init: configure the serial.

Step Eleven:

initiate the flash and nand

step twelve:

relocate the environment and then configure some data.

Step Thirteen:

The function will create a devices list and then initiate the devices,such as I2C,LCD,Video,keyboard and so on.

Step Fourteen:

The mostly devices have been initiated. Now we can open the interrupt.

Step Fifteen:

The u-boot will run a loop ,waiting a action from the user.

Note :

The “armlinux.c” file isn't referenced in initialization flow. This file will configure argument for Linux kernel, sent the argument to Linux kernel,and boot the Linux kernel.

zhaoxin@kedacom.com

2010.07.06