1. 概述

Fsl\_p2020\_ds为Freescale官方提供BSP源码，里面包含P2020的启动，基本的外设驱动。本分析报告的目的是掌握P2020的BSP启动架构，以及工程编译相关。

1. Makefile
   * 1. 顶层Makefile

# Wind River Workbench generated Makefile.

# User-specific make rules may be added in this project's "Makefile" file.

WIND\_HOME := $(subst \,/,$(WIND\_HOME)) //从环境变量中读取WIND\_HOME的值

WIND\_BASE := $(subst \,/,$(WIND\_BASE)) //同上

all : $(BUILD\_SPEC) //第一个执行目标->跳转到依赖$(BUILD\_SPEC)，即bootloader\_uncmp.bin

@echo "make 111111111111111111"

@echo "make: built targets of $(BSP\_DIR)"

BUILD\_SPEC := bootloader\_uncmp.bin

BSP\_DIR := $(CURDIR)/fsl\_p2020\_ds

BSP := fsl\_p2020\_ds

TOOL := e500v2gnu

-include \*.makefile

target\_% : $(BUILD\_SPEC)

@echo "make: built targets of $(BSP\_DIR)"

clean :

@echo "make: removing targets and objects of $(BSP\_DIR)";\

cd $(BSP\_DIR);\

$(MAKE) clean

# Make rule to build bootloader images.

# Maps all UI bootloader build targets to BSP-internal bootrom make rules,

# changes to the BSP directory, and actually builds the bootloader image.

# Valid for following user-defined build target names (bootloader styles):

# bootloader (Compressed)

# bootloader\_uncmp (Uncompressed)

# bootloader\_res (Resident)

# bootloader\_res\_high (Resident At High Address)

# Valid for following extensions (bootloader formats):

# [none] (Default (ELF) Format)

# .bin (Bin Format)

# .hex (Hex Format)

bootloade% ://Make首先执行脚本的地方

@echo "make 22222222222222222"

@target=$(subst loader,rom,$@);\

echo "building $$target image for $(BSP\_DIR)";\

cd $(BSP\_DIR);\ //进入BSP目录（即./fsl\_p2020\_ds中）

$(MAKE) $$target TOOL=$(TOOL) //执行BSP目录中的Makfile,即./fsl\_p2020\_ds/Makefile

# Make rule to build additional vxWorks images.

# Changes to the BSP directory, and actually builds the vxWorks image.

# Valid for following images:

# vxWorks

# vxWorks.sym

# vxWorks.st

# vxWorks.st\_rom

# vxWorks.res\_rom

# ... and similar

vxWork% :

@echo "building $@ image for $(BSP\_DIR)";\

cd $(BSP\_DIR);\

$(MAKE) $@ TOOL=$(TOOL)

1. 链接

ldppc -X -N -e \_romInit \ -LC:/WindRiver/vxworks-6.8/target/lib/ppc/PPC32/e500v2gnu\_standard -LC:/WindRiver/vxworks-6.8/target/lib/ppc/PPC32/e500v2gnu\_ppc85XX\_e500v2 -LC:/WindRiver/vxworks-6.8/target/lib/ppc/PPC32/e500v2common\_ppc85XX\_e500v2 -LC:/WindRiver/vxworks-6.8/target/lib/ppc/PPC32/e500v2gnu -LC:/WindRiver/vxworks-6.8/target/lib/ppc/PPC32/e500v2common -LC:/WindRiver/components/obj/vxworks-6.8/krnl/lib/ppc/PPC32/e500v2gnu -LC:/WindRiver/components/obj/vxworks-6.8/krnl/lib/ppc/PPC32/e500v2common -lipappl-basic -lipcom-basic -lipcom\_vxworks-basic -lipmcrypto-basic -lipnet-basic -lipnet\_krn-basic -liptcp-basic -lvxcompat-basic -lvxcoreip-basic -lvxmux-basic -lstl -larch -lhwdb -lcplus -lvxTestV2 -laim -lboot -lc\_wr -lcommoncc -ldcc -ldebug -ldevice -ldrv -ldshm -ldsi -lerfLib -lfp -lfsl\_spa -lgcc -lmultios\_ipc -los -lpxtrace -lrpc -lspl -lssi -ltffs -ltipc -lusb -lusb2 -lvnode -lvxmp -lwdb -lwind -lwindview -lwrload -lxbd \

-Ttext 01000000 -o bootrom\_uncmp romInit.o bootInit\_uncmp.o \

ctmp.o ctdt.o -defsym \_VX\_DATA\_ALIGN=1 -defsym wrs\_kernel\_rom\_size=0x000100000 -T C:/WindRiver/vxworks-6.8/target/h/tool/gnu/ldscripts/link.DOTBOOTRAM

链接命令主要关心两个点：一个是**入口地址**，一个是**链接地址**。入口地址表明整个bsp第一个调用的函数（或标签），而链接地址表明整个镜像被加载到RAM中时的地址。