Модуль FMC130E

Технологическое программное обеспечение

2019.10.24 Версия 1.0

1. Общие положения

Предполагается, что на модуле установлены следующие типы энергонезависимой памяти: SPI Flash (далее – QSPI), eMMC Flash (далее – eMMC) и соединитель для подключения карт microSD (далее карта microSD – SD).

В зависимости от состояния перемычки J1 “SPIO” на модуле FMC130E, начальная загрузка выполняется из QSPI или SD. В загрузочный файл BOOT.BIN включаются только fsbl.elf (FSBL) и u-boot.elf (u-boot). После того, как FSBL передаст управление u-boot, выполняется заданный сценарий загрузки системы, включая конфигурирование программируемой логики (PL) Zynq.

Содержимое загрузочного раздела QSPI:

fsbl.elf – первичный загрузчик.

u-boot.elf – вторичный загрузчик.

Содержимое загрузочного раздела BOOT (FAT32) eMMC/SD:

image.ub.sd – ядро Linux с файловой системой на SD/eMMC, используется для непосредственной работы системы.

image.ub.ramfs – ядро Linux с файловой системой INITRAMFS, используется для настройки eMMC при первичном включении, а так же для устранения проблем и восстановления системы.

system.bit.bin – конвертированный файл прошивки ПЛИС (файла конфигурации ПЛИС базового модуля). Файлов прошивки может быть много, и все они расположены в соответствующих каталогах.

system.dtb – конвертированный файл с описанием устройств программируемой логики ПЛИС базового модуля. Используется ядром Linux для загрузки соответствующих драйверов.

Установка и развертывание системы программного обеспечения

Установка petalinux-2018.2 и Xilinx SDK 2018.2

Для установки средств разработки потребуется ПО, которое можно загрузить по адресу https://www.xilinx.com/downloads. Требуемые версии пакетов: 2018.2. Данное ПО устанавливается соответственно в каталоги:

/opt/xilinx/petalinux-2018.2;

/opt/xilinx/SDK/2018.2.

Для установки ПО следуйте инструкциям программы. После установки ПО рекомендуется создать символьные ссылки на созданные директории. Это позволит работать пользовательским скриптам с несколькими версиями ПО без изменений самих скриптов (менять нужно будет только ссылку на требуемые версии petalinux и SDK).

```
cd /opt/xilinx
ln -svf petalinux-2018.2 petalinux
ln -svf SDK/2018.2 sdk
```

2. Развертывание программного обеспечения

Клонирование проекта поддержки модулей на базе процессоров Zynq:

cd ~

git clone https://github.com/insys-projects/xproject.git xproject

Сборка целевой системы:

cd xproject/scripts

source /opt/xilinx/petalinux/settings.sh

Затем необходимо запустить скрипт makeproject.sh, передав ему в качестве параметра имя директории, где находится hdf-файл описания аппаратуры. Все hdf-файлы расположены в подкаталогах первого уровня директории xproject. Имена подкаталогов сформированы по следующему правилу: <имя модуля>_<версия модуля>_<версия микросхемы>_<субмодуль>. Например: **fmc130e_v12_z045_fmctest**:

./makeproject ../fmc130e_v12_z045_fmctest

В указанном каталоге, скриптом будет создан подкаталог проекта с именем project. После этого последовательно будет выполнена конфигурация проекта petalinux, файловой системы (rootfs) и ядра Linux (kernel). В последовательно открывающихся графических меню. После конфигурации запустится сборка всех компонентов проекта. После окончания сборки все требуемые для дальнейшей работы компоненты будут находиться в каталоге:

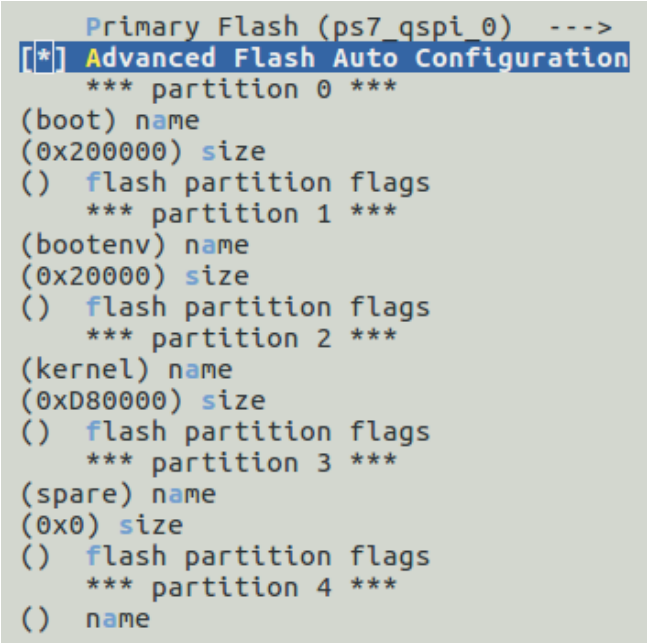
~/xproject/fmc130e_v12_z045_fmctest/project/images/linux

Используя скрипт liteproject.sh можно сгенерировать проект без конфигурации kernel и rootfs. Это ускорит создание проекта, а настройку можно выполнить позднее.

./liteproject ../fmc130e_v12_z045_fmctest

Откроется диалоговое окно, в котором можно выполнить требуемую настройку параметров системы. Настраиваем следующие пункты меню, остальные оставляем по умолчанию:

Subsystem AUTO Hardware Settings → Flash Settings



Subsystem AUTO Hardware Settings → Advanced bootable images storage Settings →

dtb image settings →

image storage media (primary sd)

Image Packaging Configuration →

Root filesystem type (SD card)

(/dev/mmcblk0p2) Device node of SD device (NEW)

(image.ub) name for bootable kernel image

(0x1000) DTB padding size

[ ] Copy final images to tftpboot

Firmware Version Configuration →

(fmc130e) Host name

(fmctest) Product name

(1.2) Firmware Version

Конфигурация корневой файловой системы rootfs:

Filesystem Packages → misc → gcc-runtime

[*] libstdc++

Filesystem Packages → base → e2fsprogs

[*] e2fsprogs

[*] e2fsprogs-resize2fs

[*] e2fsprogs-badblocks

[*] e2fsprogs-e2fsck

[*] e2fsprogs-tune2fs

[*] e2fsprogs-mke2fs

Filesystem Packages → base → util-linux

[*] util-linux

[*] util-linux-mkfs

[*] util-linux-mount

[*] util-linux-fdisk

[*] util-linux-fsck

Filesystem Packages → base → tar

[*] tar

Filesystem Packages → console → utils → mc

[*] mc

Filesystem Packages → console → utils → bzip

[*] bzip2

После настройки нужно выйти из меню с сохранением всех изменений и дождаться окончания конфигурации проекта.

Конфигурация ядра Linux:

Device Drivers →

<*> NVM Express block device

После настройки выходим из меню с сохранением всех изменений и ждем окончания сборки проекта.

Конвертирование файла ядра Linux для раздельной загрузки DTB и монтированием файловой системы на SD или eMMC

Для обеспечения загрузки ядра Linux с использованием отдельного файла system.dtb требуется выполнить следующие команды:

cd ~/xproject/fmc130e_v12_z045_fmctest/project/images/linux

cp ~/xproject/common/kernel/image-2018.2-kernel.its ./

cp ~/xproject/common/kernel/image.sh ./

./image.sh image-2018.2-kernel.its

После этого в текущем каталоге будет создан конвертированный файл с именем image.ub. Этот файл требуется переименовать в image.ub.sd и поместить на загрузочный раздел SD карты.

Конфигурация проекта petalinux с загрузкой из INITRAMFS

На данном этапе настройки необходимо повторно выполнить конфигурацию проекта, выполнив следующие команды:

cd ~/xproject/fmc130e_v12_z045_fmctest/project

petalinux-config

Image Packaging Configuration →

Root filesystem type (INITRAMFS) --->

Остальное по умолчанию. Выйти с сохранением новых настроек.

Выполнить сборку командой**:**

petalinux-build

После окончания сборки, необходимо выполнить операции из следующего раздела.

Конвертирование файла ядра Linux для раздельной загрузки DTB и монтированием файловой системы в RAM

Для обеспечения загрузки ядра Linux с использованием отдельного system.dtb и монтированием файловой системы в RAM требуется выполнить следующие команды:

cd ~/xproject/fmc130e_v12_z045_fmctest/project/images/linux

cp ~/xproject/common/kernel/image-2018.2-kernel.its ./

cp ~/xproject/common/kernel/image.sh ./

./image.sh image-2018.2-kernel.its

После этого в текущем каталоге будет создан конвертированный файл с именем **image.ub**. Этот файл требуется переименовать в **image.ub.ramfs** и поместить его на загрузочный раздел SD карты.

Конвертирование файла для загрузки программируемой логики из SD

Для обеспечения загрузки программируемой логики новым файлом конфигурации требуется выполнить конвертирование файла system.bit. Для этого необходимо выполнить следующие команды:

cd ~/xproject/fmc130e_v12_z045_fmctest/project/images/linux

cp ~/xproject/common/qspi/bitconv.bif ./

cp ~/xproject/common/qspi/bitconv.sh ./

./bitconv.sh

После этого в текущем каталоге будет создан файл с именем **system.bit.bin**. Этот файл требуется поместить на загрузочный раздел SD карты.

Создание файла BOOT.BIN для первичной загрузки из SD и файла qspi_zynq.bin для записи в QSPI

Для обеспечения загрузки из SD на загрузочном разделе должен находиться файл BOOT.BIN. Этот файл содежит в себе первичный загрузчик fsbl и вторичный загрузчик u-boot. Для того, чтобы получить файл BOOT.BIN необходимо выполнить следующие команды:

cd ~/xproject/fmc130e_v12_z045_fmctest/project/images/linux

cp ~/xproject/common/qspi/qspi_zynq.bif ./

cp ~/xproject/common/qspi/qspi_zynq.sh ./

./qspi_zynq.sh

cp qspi_zynq.bin BOOT.BIN

После этого в текущем каталоге будет создано два файла. Один с именем **qspi_zynq.bin**, второй с именем **BOOT.BIN**.

Скопировать файлы **system.dtb, qspi_zynq.bin, BOOT.BIN** на загрузочный раздел SD карты.

Инициализация eMMC

Для инициализации eMMC необходимы следующие файлы:

system.bit.bin – файл конфигурации программируемой логики;

BOOT.BIN – образ первичного и вторичного загрузчика для SD/eMMC;

qspi_zynq.bin – образ первичного и вторичного загрузчика для QSPI;

rootfs.tar.bz2 – файл с архивом корневой файловой системы;

image.ub.sd – образ ядра Linux с монтированием файловой системы на SD/eMMC;

image.ub.ramfs – образ ядра Linux с файловой системой;

system.dtb – двоичный файл описания устройств (DTB).

Порядок действий для выполнения первичной настройки FMC130E

1) На FMC130E устанавлить режим загрузки из SD

2) Включить питание FMC130E

3) На HOST запустить терминальную программу minicom

sudo minicom -D /dev/ttyUSB1 -b 115200

4) Выполнить выключение FMC130E и последующее включение, через 5-10 секунд.

5) В окне minicom нажать ESC, чтобы остановить загрузку системы на этапе u-boot.

6) Добавить переменные среды

Zynq> setenv SN 19.07.7287.84717

Zynq> setenv PID 84717

Zynq> setenv ethaddr 00:1c:77:01:4A:ED

7) Выполнить запись QSPI в окне u-boot

#run sd_on; run burnspi

8) Выполнить загрузку ПЛИС и загрузить образ image.ub.ramfs системы, для разметки файловой системы на EMMC.

#run sd_on; run loadbit; run loaddtb; run loadimgr; run emmc_on;

#bootm 0x10000000 - 0x20000000;

8) После загрузки Linux, выполнить вход в систему. Логин: root, пароль: root. Затем выполнить разметку eMMC командой fdsik:

root@fmc130e:~# fdisk /dev/mmcblk0

Welcome to fdisk (util-linux 2.30).

Changes will remain in memory only, until you decide to write them.

Be careful before using the write command.

Command (m for help): n

Partition type

p primary (0 primary, 0 extended, 4 free)

e extended (container for logical partitions)

Select (default p): p

Partition number (1-4, default 1): 1

First sector (2048-28835839, default 2048):

Last sector, +sectors or +size{K,M,G,T,P} (2048-28835839, default 28835839): +1G

Created a new partition 1 of type 'Linux' and of size 1 GiB.

Partition #1 contains a vfat signature.

Do you want to remove the signature? [Y]es/[N]o: Y

The signature will be removed by a write command.

Command (m for help): n

Partition type

p primary (1 primary, 0 extended, 3 free)

e extended (container for logical partitions)

Select (default p): p

Partition number (2-4, default 2):

First sector (2099200-28835839, default 2099200):

Last sector, +sectors or +size{K,M,G,T,P} (2099200-28835839, default 28835839):

Created a new partition 2 of type 'Linux' and of size 12.8 GiB.

Partition #2 contains a ext4 signature.

Do you want to remove the signature? [Y]es/[N]o: Y

The signature will be removed by a write command.

Command (m for help): w

The partition table has been altered.

Calling ioctl() to re-read parti mmcblk0: p1 p2

tion table.

Syncing disks.

9) Создать файловые системы на разделах EMMC

mkfs.vfat -n BOOT /dev/mmcblk0p1

mkfs.ext4 -L rootfs /dev/mmcblk0p2

10) Выполнить перезагрузку системы и остановить загрузку в u-boot;

11) Выполнить копирование данных из SD на EMMC

#run copyall

12) Выполнить загрузку образа image.ub.ramfs

#run sd_on; run loadbit; run loadimgr; run emmc_on; bootm 0x10000000;

13) Выполнить вход в систему – логин: root, пароль: root. Затем выполнить распаковку образа с файловой системой на второй раздел EMMC:

#cd /run/media/mmcblk0p1

#tar -xjvf rootfs.tar.bz2 -C /run/media/mmcblk0p2/

14) Выполнить перезагрузку системы и остановить загрузку в u-boot;

15) Настроить загрузку системы из eMMC

#setenv bootcmd 'run emmc_on; run start;'

#saveenv;

16) Выключить питание FMC130E и вынуть SD. Для FMC130E установить режим первичной загрузки из QSPI. Включить питание. Система должна выполнить старт из QSPI и последующую загрузку из eMMC.

3. Работа с библиотекой BARDY

Для разработки пользовательских приложений используется библиотека bardy. Для модулей на базе процессоров Zynq имеются следующие компоненты:

libbrd.so – пользовательский API

libbzynq.so – базовый драйвер для работы с устройством

lzynq.ko – модуль ядра для обеспечения доступа к аппаратуре, обработке прерываний, работы с каналами DMA.

Библиотека bardy использует для своей работы вспомогательную бибиотеку gipcy для обеспечения коссплатформенного IPC (Inter Process Communication). Библиотека gipcy состоит из следующих компонент:

libbgipcy.so – библиотека IPC

ipcdrv.ko – модуль ядра для обеспечения работы libbgipcy.so.

Клонировать вспомогательную библиотеку gipcy для поддержки IPC

```
cd ~
git clone https://github.com/insys-projects/gipcy.git  gipcy.git
export GIPCYDIR=~/gipcy.git
```

Распаковать библиотеку bardy

```
tar -xzvf XXXX-LS.tgz -C  bardy
```

Настроить переменных окружения для кросскомпиляции

source /opt/xilinx/petalinux/settings.sh

cd ~/xproject/scripts

source ./envarm.sh ../fmc130e_v12_z045_fmctest

cd ~/bardy

source ./BardyEnv.sh

Сборка библиотеки GIPCY

cd $GIPCYDIR

make clean && make

Сборка библиотеки BARDY

cd ~/bardy

make clean && make

После окончания сборки в каталоге bardy/bin должны быть созданы двоичные файлы всех компонент, поставляемых в исходных текстах в составе библиотеки bardy.

Сборка примера exam_adc из состава BARDY

cd ~/bardy/EXAM/exam_adc

make clean && make

Описание команд u-boot используемых при настройке модуля

fileaddr=0x10000000 – адрес в памяти для загрузки файлов;
dtbaddr=0x20000000 – адрес в памяти для загрузки DTS;
sel_pin=48 – номер вывода для выбора активного носителя SD или eMMC;
sd_on – установить активным носителем SD;
emmc_on – установить активным носителем eMMC;
burnspi – записать файл qspi_zynq.bin из активного носителя в QSPI;
loadfile – загрузить файл с активного носителя;
savefile – сохранить файл на активный носитель;
copyfile – скопировать файл с SD на eMMC;
copyldr – скопировать файл BOOT.BIN с SD на eMMC;
copydtb – скопировать файл system.dtb с SD на eMMC;
copybit – скопировать файл system.bit.bin с SD на eMMC;
copyimg – скопировать файл image.ub.sd с SD на eMMC;
copyimgr – скопировать файл image.ub.ramfs с SD на eMMC;
copyfs – скопировать файл rootfs.tar.bz2 с SD на eMMC;
loadbit – загрузить файл конфигурации программируемой логики ПЛИС;
loadimg – загрузить файл image.ub.sd в память PS;
loaddtb – загрузить файл system.dtb в память PS;
loadimgr – загрузить файл image.ub.ramfs в память PS;
start – выполнить загрузку системы с файловой системой на SD/eMMC;
startr – выполнить загрузку системы с файловой системой INITRAMFS;
bootcmd – активный режим загрузки;

Остановить загрузку системы можно нажав клавишу ESC на этапе работы u-boot в программе minicom. Для загрузки системы, u-boot считывает переменные среды из QSPI и выполняет обработку переменной bootcmd.

Например:

bootcmd = run sd_on; run start; - активным носителем будет выбрана SD-карта и затем обработана команда start.

Значение всех переменных u-boot можно вывести командой: print. Интерактивную справку по всем командам u-boot можно получить командой: help. Установить переменную u-boot можно командой setenv. Например: setenv bootcmd ‘run emmc_on; run start;’ Сохранить все переменные среды в QSPI можно командой saveenv.