

Портирование HAL eCos и загрузчика RedBoot на микропроцессоры «Эльбрус-2С3» для бортовых встраиваемых систем

Калинин Михаил Юрьевич, АО НТЦ «Модуль»

Титов Юрий Павлович ФГАОУ ВО «Московский авиационный институт (МАИ)», г. Москва, Волоколамское ш., д4

Актуальность для авиационно-космических встраиваемых систем

В бортовых цифровых вычислительных машинах авиации и космоса традиционно используются загрузчики с ограниченной функциональностью, часто привязанные к конкретному BIOS. Платформа «Эльбрус-2С3» предлагает высокую производительность на VLIW-архитектуре, но штатная программа начального старта не обеспечивает сервисов, критически важных при разработке и эксплуатации бортового ПО: сетевую загрузку образов по TFTP, интерактивную работу с Flash-памятью, встроенный отладчик GDB stub для удалённой отладки, поддержку протоколов X/Y-modem для полевых обновлений прошивки. Загрузчик RedBoot, построенный на основе OCPB eCos, предоставляет все перечисленные возможности. Однако архитектура e2k процессора «Эльбрус-2С3» (VLIW, оконный стек регистров, программное управление TLB, двухъядерность с нестандартной когерентностью кэшей) не поддерживается в существующей кодовой базе RedBoot.

Архитектурные особенности «Эльбрус-2С3» как вызов для HAL eCos

HAL eCos имеет трехуровневую структуру (Architecture, Variant, Platform). Для «Эльбрус-2С3» требуется создание архитектурного HAL «с нуля». Регистровая модель процессора включает 256 регистров общего назначения, из которых 224 организованы в оконный стек процедур, а 32 являются глобальными. При переключении контекста необходимо сохранять не только указатель текущего окна (CWP), но и регистры связующей информации (PCS), содержащие адрес возврата и признаки VLIW-упаковки инструкций. Предлагаемая структура HAL_SavedRegisters включает поля для всех указанных компонентов, что обеспечивает корректное восстановление потока команд после прерывания или системного вызова. Контроллер APIC в «Эльбрус-2С3» поддерживает распределение прерываний между двумя ядрами. Разрабатываемый диспетчер на ассемблере сохраняет состояние VLIW-конвейера, определяет номер активного ядра через регистр конфигурации, маскирует прерывания на данном ядре и передаёт управление стандартному обработчику eCos. Архитектура e2k не имеет аппаратного TLB с фиксированным набором записей; трансляция адресов выполняется через программно загружаемые регистры РТСП. Макросы HAL_MMU_* реализуют прямую запись в эти регистры, а не используют стандартный механизм кэширования eCos, рассчитанный на классические RISC. Среда разработки «Эльбрус» использует компилятор LCC (EDG с синтаксисом встроенного ассемблера, отличным от GCC. Для обеспечения переносимости применяется условная компиляция через макросы __e2k__ и __LCC__, а также обход устаревшей поддержки C++ в LCC путём ограничения кода RedBoot подмножеством C99.

Методология портирования RedBoot для БЦВМ на базе «Эльбрус-2С3»

Портирование выполняется в два этапа с промежуточной отладкой в QEMU. На первом этапе создаётся платформенный HAL для целевой платы (например, сетевого контроллера интерфейсов). За основу берётся шаблон MIPS, но с полной заменой ассемблерных модулей. В CDL-скриптах задаются физические адреса: Flash (0x00000000), RAM (0x80000000), области отображения периферии (UART, APIC, таймер). Драйвер последовательного порта реализуется в режиме опроса (polling), так как RedBoot на начальном этапе не использует прерывания. Функции `cyg_hal_plf_serial_putc/getc` работают непосредственно с регистрами FIFO UART, адреса которых уточняются по документации на системный контроллер «Эльбрус-2С3». На втором этапе выполняется сборка RAM-версии RedBoot (ELF-образ), которая загружается через JTAG-отладчик поверх штатного BIOS. Это позволяет отладить

инициализацию APIC, системного таймера и базовый ввод-вывод без риска повреждения Flash. После стабилизации настраивается ROM-конфигурация: образ линкуется на адрес 0x0, таблица векторов прерываний размещается по физическому адресу сброса процессора. Для прошивки используется модифицированная версия flashrom с поддержкой протокола SPI, адаптированного под чип Flash на целевой плате. Штатный BIOS замещается полностью: после включения питания управление передаётся RedBoot, что позволяет инженеру-разработчику бортовой аппаратуры сразу получить командную строку и сервисы отладки.

Адаптация GDB stub и отладка в QEMU для двухъядерной архитектуры

RedBoot включает встроенный GDB stub, позволяющий отлаживать загруженные приложения непосредственно в бортовой вычислительной системе. При активации CYGDBG_HAL_DEBUG_GDB_INCLUDE_STUBS возникает конфликт: stub пытается включить прерывания UART, но RedBoot в eCos не имеет их поддержки. Предлагаемое решение — переопределение макросов в hal_diag.h, замена прерываний на цикл опроса флага готовности приёмника внутри обработчика GDB-пакетов. Для ускорения отладки используется эмулятор QEMU. В рамках работы предлагается доработка qemu-system-e2k для запуска bare-metal кода на модели платы «Эльбрус-2С3». Необходимо добавить описание машины (-M elbrus_2s3), эмулирующей два ядра, контроллер прерываний APIC с распределением прерываний, системный таймер и UART. Эмуляция Flash реализуется через бэкэнд m25p80 с параметрами, соответствующими реальному чипу. Образ redboot.bin подаётся через -bios. Встроенный GDB stub эмулятора позволяет отлаживать код инициализации (настройку TLB, обработку первого исключения) на уровне VLIW-инструкций, что на реальном оборудовании крайне затруднительно.

Учёт двухъядерности и когерентности кэшей в интересах авиационных систем

Процессор «Эльбрус-2С3» имеет два ядра с разделёнными кэшами L1 и общей кэш-памятью L2, но с программно-управляемой когерентностью. Для бортовых систем, где часто применяется резервирование и режим «одно ядро в работе, второе в горячем резерве», требуется корректно инициализировать только одно ядро, а второе удерживать в сбросе. В HAL вводится функция hal_cpu_startup, которая по значению регистра идентификатора ядра определяет, какое ядро выполняет код. Первичное ядро (ядро 0) проходит полную инициализацию и запускает RedBoot. Вторичное ядро (ядро 1) переводится в бесконечный цикл с отключёнными прерываниями. Для когерентности кэшей на этапе загрузчика используется простейшая стратегия, приемлемая для авиационного ПО с детерминированным выполнением: при любых операциях DMA (например, при загрузке конфигурации сетевого контроллера по TFTP) выполняется сброс (invalidate) кэш-линий для диапазона адресов через специальную инструкцию cflush.

Заключение: применимость в бортовых вычислителях и сетевых контроллерах

Предлагаемая методология портирования RedBoot на «Эльбрус-2С3» позволяет создать унифицированную среду начальной загрузки и отладки для БЦВМ авиационного и космического назначения. В результате разработчик получает инструменты, стандартные для мировой практики (TFTP, GDB, Flash Image System), но работающие на отечественной VLIW-архитектуре. Замена штатного BIOS на RedBoot сокращает время отладки бортового ПО, упрощает полевое обновление прошивки и повышает наблюдаемость системы на ранних этапах включения питания. Дальнейшие работы включают портирование полного ядра eCos с поддержкой многопоточности на обоих ядрах и реализацию драйверов бортовых интерфейсов поверх созданного HAL.