An ARM from shoulder to hand

Soeren Straarup xride@FreeBSD.ORG

September 14, 2007

What is ARM?

- Is it a CPU or a philosopy?
- It is more or less an idea to how to make a RISC computer
- Focus is on the molularness and low power usage
- One should be able to mass produce ARM processors on older fabs.
- Has many variations still active (ARM7, ARM9, ..)

What is ARM?

- Is it a CPU or a philosopy?
- It is more or less an idea to how to make a RISC computer
- Focus is on the molularness and low power usage
- One should be able to mass produce ARM processors on older fabs.
- Has many variations still active (ARM7, ARM9, ..)

What is ARM?

- Is it a CPU or a philosopy?
- It is more or less an idea to how to make a RISC computer
- Focus is on the molularness and low power usage
- One should be able to mass produce ARM processors on older fabs.
- Has many variations still active (ARM7, ARM9, ..)

What is ARM?

- Is it a CPU or a philosopy?
- It is more or less an idea to how to make a RISC computer
- Focus is on the molularness and low power usage
- One should be able to mass produce ARM processors on older fabs.
- Has many variations still active (ARM7, ARM9, ..)

What is ARM?

- Is it a CPU or a philosopy?
- It is more or less an idea to how to make a RISC computer
- Focus is on the molularness and low power usage
- One should be able to mass produce ARM processors on older fabs.
- Has many variations still active (ARM7, ARM9, ..)

What is ARM?

- Is it a CPU or a philosopy?
- It is more or less an idea to how to make a RISC computer
- Focus is on the molularness and low power usage
- One should be able to mass produce ARM processors on older fabs.
- Has many variations still active (ARM7, ARM9, ..)

How to support the many variations of boards?

- Getting the DUT to boot
- With Atmels AT91RM9200 there is an internal boot loader, which can recieve via xmodem over a serial connection.
- But how much internal RAM?
- It goes for AT91RM9200 since it got about 12KB of internal RAM where as one can use about 10KB And the current bootiic bootloader is about 9KB big.
- Abstraction Layers
- Since There are multiple kinds of cores and multiple ways of wrapping it.

How to support the many variations of boards?

- Getting the DUT to boot
- With Atmels AT91RM9200 there is an internal boot loader, which can recieve via xmodem over a serial connection.
- But how much internal RAM?
- It goes for AT91RM9200 since it got about 12KB of internal RAM where as one can use about 10KB And the current bootiic bootloader is about 9KB big.
- Abstraction Layers
- Since There are multiple kinds of cores and multiple ways of wrapping it.

How to support the many variations of boards?

- Getting the DUT to boot
- With Atmels AT91RM9200 there is an internal boot loader, which can recieve via xmodem over a serial connection.
- But how much internal RAM?
- It goes for AT91RM9200 since it got about 12KB of internal RAM where as one can use about 10KB And the current bootiic bootloader is about 9KB big.
- Abstraction Layers
- Since There are multiple kinds of cores and multiple ways of wrapping it.

How to support the many variations of boards?

- Getting the DUT to boot
- With Atmels AT91RM9200 there is an internal boot loader, which can recieve via xmodem over a serial connection.
- But how much internal RAM?
- It goes for AT91RM9200 since it got about 12KB of internal RAM where as one can use about 10KB And the current bootiic bootloader is about 9KB big.
- Abstraction Layers
- Since There are multiple kinds of cores and multiple ways of wrapping it.

How to support the many variations of boards?

- Getting the DUT to boot
- With Atmels AT91RM9200 there is an internal boot loader, which can recieve via xmodem over a serial connection.
- But how much internal RAM?
- It goes for AT91RM9200 since it got about 12KB of internal RAM where as one can use about 10KB And the current bootiic bootloader is about 9KB big.
- Abstraction Layers
- Since There are multiple kinds of cores and multiple ways of wrapping it.

How to support the many variations of boards?

- Getting the DUT to boot
- With Atmels AT91RM9200 there is an internal boot loader, which can recieve via xmodem over a serial connection.
- But how much internal RAM?
- It goes for AT91RM9200 since it got about 12KB of internal RAM where as one can use about 10KB And the current bootiic bootloader is about 9KB big.
- Abstraction Layers
- Since There are multiple kinds of cores and multiple ways of wrapping it.

Atmel AT91RM9200

- It has no kind of internal display drivers
- It is basicly back to basics
- Though it has an ethernet MAC periphial but no PHY
- This leads to the modular design of software
- Where a network interface is not just one unit, but consists of different parts

Atmel AT91RM9200

- It has no kind of internal display drivers
- It is basicly back to basics
- Though it has an ethernet MAC periphial but no PHY
- This leads to the modular design of software
- Where a network interface is not just one unit, but consists of different parts

Atmel AT91RM9200

- It has no kind of internal display drivers
- It is basicly back to basics
- Though it has an ethernet MAC periphial but no PHY
- This leads to the modular design of software
- Where a network interface is not just one unit, but consists of different parts

Atmel AT91RM9200

- It has no kind of internal display drivers
- It is basicly back to basics
- Though it has an ethernet MAC periphial but no PHY
- This leads to the modular design of software
- Where a network interface is not just one unit, but consists of different parts

Atmel AT91RM9200

- It has no kind of internal display drivers
- It is basicly back to basics
- Though it has an ethernet MAC periphial but no PHY
- This leads to the modular design of software
- Where a network interface is not just one unit, but consists of different parts

First go at the code

- Setup a buildenv
- One way is to
- Make sure that /usr/src is there

First go at the code

- Setup a buildenv
- One way is to
- Make sure that /usr/src is there

First go at the code

- Setup a buildenv
- One way is to
- Make sure that /usr/src is there

First go at the code

- Setup a buildenv
- One way is to
- Make sure that /usr/src is there

Source, important files

- cd /path/to/src/sys
- conf/options.arm To define options that can be used in the KERNCONF file
- boot/arm/at91 This is the place for the boot code for Atmel board
- boot/arm/at91/Makefile.inc Is interesting with regards to variable BOOT_FLAVOR.
  Which defines what board the bootloader should be build for.
- *libat*91 / *arm _ init*.*S* This file holds the automated boot menu sequence.
  That are mac address, local ip, tftpservers ip, what to tftp and what to execute.
- make TARGET=arm *TARGET _ ARCH = arm buildenv*
- cd sys/boot/arm/at91
- make

Source, important files

- cd /path/to/src/sys
- conf/options.arm To define options that can be used in the KERNCONF file
- boot/arm/at91 This is the place for the boot code for Atmel board
- boot/arm/at91/Makefile.inc Is interesting with regards to variable BOOT_FLAVOR.
  Which defines what board the bootloader should be build for.
- *libat*91 / *arm* _ *init*.*S* This file holds the automated boot menu sequence.
  That are mac address, local ip, tftpservers ip, what to tftp and what to execute.
- make TARGET=arm *TARGET* _ *ARCH* = *arm buildenv*
- cd sys/boot/arm/at91
- make

Source, important files

- cd /path/to/src/sys
- conf/options.arm To define options that can be used in the KERNCONF file
- boot/arm/at91 This is the place for the boot code for Atmel board
- boot/arm/at91/Makefile.inc Is interesting with regards to variable BOOT_FLAVOR.
  Which defines what board the bootloader should be build for.
- *libat*91 / *arm _ init*.*S* This file holds the automated boot menu sequence.
  That are mac address, local ip, tftpservers ip, what to tftp and what to execute.
- make TARGET=arm *TARGET _ ARCH = arm buildenv*
- cd sys/boot/arm/at91
- make

Source, important files

- cd /path/to/src/sys
- conf/options.arm To define options that can be used in the KERNCONF file
- boot/arm/at91 This is the place for the boot code for Atmel board
- boot/arm/at91/Makefile.inc Is interesting with regards to variable BOOT_FLAVOR.
  Which defines what board the bootloader should be build for.
- *libat91 / arm _ init.S* This file holds the automated boot menu sequence.
  That are mac address, local ip, tftpservers ip, what to tftp and what to execute.
- make TARGET=arm *TARGET _ ARCH = arm buildenv*
- cd sys/boot/arm/at91
- make

Source, important files

- cd /path/to/src/sys
- conf/options.arm To define options that can be used in the KERNCONF file
- boot/arm/at91 This is the place for the boot code for Atmel board
- boot/arm/at91/Makefile.inc Is interesting with regards to variable BOOT_FLAVOR.
  Which defines what board the bootloader should be build for.
- *libat*91 / *arm* _ *init*.*S* This file holds the automated boot menu sequence.
  That are mac address, local ip, tftpservers ip, what to tftp and what to execute.
- make TARGET=arm *TARGET* _ *ARCH* = *arm buildenv*
- cd sys/boot/arm/at91
- make

Source, important files

- cd /path/to/src/sys
- conf/options.arm To define options that can be used in the KERNCONF file
- boot/arm/at91 This is the place for the boot code for Atmel board
- boot/arm/at91/Makefile.inc Is interesting with regards to variable BOOT_FLAVOR.
  Which defines what board the bootloader should be build for.
- *libat*91 / *arm* _ *init*.*S* This file holds the automated boot menu sequence.
  That are mac address, local ip, tftpservers ip, what to tftp and what to execute.
- make TARGET=arm *TARGET* _ *ARCH* = *arm buildenv*
- cd sys/boot/arm/at91
- make

Source, important files

- cd /path/to/src/sys
- conf/options.arm To define options that can be used in the KERNCONF file
- boot/arm/at91 This is the place for the boot code for Atmel board
- boot/arm/at91/Makefile.inc Is interesting with regards to variable BOOT_FLAVOR.
  Which defines what board the bootloader should be build for.
- *libat*91 / *arm* _ *init*.*S* This file holds the automated boot menu sequence.
  That are mac address, local ip, tftpservers ip, what to tftp and what to execute.
- make TARGET=arm *TARGET* _ *ARCH* = *arm buildenv*
- cd sys/boot/arm/at91
- make

Source, important files

- cd /path/to/src/sys
- conf/options.arm To define options that can be used in the KERNCONF file
- boot/arm/at91 This is the place for the boot code for Atmel board
- boot/arm/at91/Makefile.inc Is interesting with regards to variable BOOT_FLAVOR.
  Which defines what board the bootloader should be build for.
- *libat*91 / *arm* _ *init*.*S* This file holds the automated boot menu sequence.
  That are mac address, local ip, tftpservers ip, what to tftp and what to execute.
- make TARGET=arm *TARGET* _ *ARCH* = *arm buildenv*
- cd sys/boot/arm/at91
- make

Source, important files

- cd /path/to/src/sys
- conf/options.arm To define options that can be used in the KERNCONF file
- boot/arm/at91 This is the place for the boot code for Atmel board
- boot/arm/at91/Makefile.inc Is interesting with regards to variable BOOT_FLAVOR.
  Which defines what board the bootloader should be build for.
- *libat*91 / *arm* _ *init*.*S* This file holds the automated boot menu sequence.
  That are mac address, local ip, tftpservers ip, what to tftp and what to execute.
- make TARGET=arm *TARGET* _ *ARCH* = *arm buildenv*
- cd sys/boot/arm/at91
- make

First go at the loader build

- cd /path/to/src
- make TARGET=arm TARGET_ARCH=arm buildenv
- cd sys/boot/arm/at91
- make

First go at the loader build

- cd /path/to/src
- make TARGET=arm TARGET_ARCH=arm buildenv
- cd sys/boot/arm/at91
- make

First go at the loader build

- cd /path/to/src
- make TARGET=arm TARGET_ARCH=arm buildenv
- cd sys/boot/arm/at91
- make

First go at the loader build

- cd /path/to/src
- make TARGET=arm TARGET_ARCH=arm buildenv
- cd sys/boot/arm/at91
- make

First go at the loader build

- cd /path/to/src
- make TARGET=arm TARGET_ARCH=arm buildenv
- cd sys/boot/arm/at91
- make

How did it go

- The other committers has done a great job!
- The board was relativly easy to get to boot, using xmodem via serial debug port.
- There is a problem with the random device when that is enabled it won't boot at all.
- Next thing is to get the different hardware devices to work.

How did it go

- The other committers has done a great job!
- The board was relativly easy to get to boot, using xmodem via serial debug port.
- There is a problem with the random device when that is enabled it won't boot at all.
- Next thing is to get the different hardware devices to work.

How did it go

- The other committers has done a great job!
- The board was relativly easy to get to boot, using xmodem via serial debug port.
- There is a problem with the random device when that is enabled it won't boot at all.
- Next thing is to get the different hardware devices to work.

How did it go

- The other committers has done a great job!
- The board was relativly easy to get to boot, using xmodem via serial debug port.
- There is a problem with the random device when that is enabled it won't boot at all.
- Next thing is to get the different hardware devices to work.

How did it go

- The other committers has done a great job!
- The board was relativly easy to get to boot, using xmodem via serial debug port.
- There is a problem with the random device when that is enabled it won't boot at all.
- Next thing is to get the different hardware devices to work.

Wishlist

- Ware leveling for FS's via GEOM?
- A more uniform way of defining new boards

Wishlist

- Ware leveling for FS's via GEOM?
- A more uniform way of defining new boards

Wishlist

- Ware leveling for FS's via GEOM?
- A more uniform way of defining new boards

Questions

- Any questions?

Questions

- Any questions?

VEOF