SPI NOR framework

Part I - Why do we need this framework?

SPI bus controllers (drivers/spi/) only deal with streams of bytes; the bus controller operates agnostic of the specific device attached. However, some controllers (such as Freescale's QuadSPI controller) cannot easily handle arbitrary streams of bytes, but rather are designed specifically for SPI NOR.

In particular, Freescale's QuadSPI controller must know the NOR commands to find the right LUT sequence. Unfortunately, the SPI subsystem has no notion of opcodes, addresses, or data payloads; a SPI controller simply knows to send or receive bytes (Tx and Rx). Therefore, we must define a new layering scheme under which the controller driver is aware of the opcodes, addressing, and other details of the SPI NOR protocol.

Part II - How does the framework work?

This framework just adds a new layer between the MTD and the SPI bus driver. With this new layer, the SPI NOR controller driver does not depend on the m25p80 code anymore.

Before this framework, the layer is like:

```
          MTD
------------------------
         m25p80
------------------------
      SPI bus driver
------------------------
       SPI NOR chip
```

After this framework, the layer is like:

```
          MTD
------------------------
     SPI NOR framework
------------------------
         m25p80
------------------------
      SPI bus driver
------------------------
      SPI NOR chip
```

With the SPI NOR controller driver (Freescale QuadSPI), it looks like:

```
          MTD
------------------------
     SPI NOR framework
------------------------
       fsl-quadSPI
------------------------
      SPI NOR chip
```

Part III - How can drivers use the framework?

The main API is spi_nor_scan(). Before you call the hook, a driver should initialize the necessary fields for spi_nor{}. Please see drivers/mtd/spi-nor/spi-nor.c for detail. Please also refer to spi-fsl-qspi.c when you want to write a new driver for a SPI NOR controller. Another API is spi_nor_restore(), this is used to restore the status of SPI flash chip such as addressing mode. Call it whenever detach the driver from device or reboot the system.