Lecture 11
Flash Programming & TLV Structure

Texas Instruments Incorporated
University of Beira Interior (PT)

Pedro Dinis Gaspar, António Espírito Santo, Bruno Ribeiro, Humberto Santos
University of Beira Interior, Electromechanical Engineering Department
www.msp430.ubi.pt
Contents (1/2)

- **Flash introduction**
- **Flash memory operation and segmentation**
- **Write/erase modes**
- **Access during write/erase**
- **Flash memory controller registers**
TLV introduction

Supported Tags

Calculating the Checksum of SegmentA

Parsing the TLV Structure of SegmentA
Memory in general is broadly classified as read-only memory (ROM) or random-access memory (RAM);

Flash memory is a hybrid of ROM and RAM;

Flash memory is:
- Low cost;
- Electrically programmable;
- Fast to read from;
- High density;
- Reliable;
- Non-volatile.
MSP430Fxxx(x) flash memory structure is:
- Divided into segments;
- Allows bit-, byte- and word- addressing and programming;
- Must be erased in segments.

Flash memory controller:
- Controls programming and erase operations;
- Has 3 or 4 registers (see the device-specific data sheet);
- Has a timing generator:
  - Sourced from ACLK, SMCLK, or MCLK;
  - Flash timing generator operating frequency: 
    \( \sim 257 \text{ kHz} < f(\text{FTG}) < \sim 476 \text{ kHz} \) (see device-specific data);
  - The selected clock source should be divided using the FNx bits to meet the frequency requirements of \( f(\text{FTG}) \).
Flash memory controller:

- Timing generator block diagram:

- Uses a voltage generator to supply programming and erase voltages. The output voltage must be stable.
An MSP430 flash device can be programmed via:

- JTAG interface (requires four signals, ground and optionally VCC and RST/NMI);
- Bootstrap Loader (using a UART serial interface);
- Custom solution (using one of the interfaces available and through user developed software).
- **MSP430 flash memory block diagram:**
  - MSP430FG4618 (Experimenter’s board) has two flash memory arrays.
The flash memory partitions (device-specific data):

- Main memory section (two or more 512-byte segments);
- Information memory section (two 128-byte segments), located at lower memory addresses, in the address space immediately following RAM.
2xx family: SegmentA (information A):

- eZ430-F2013;
- eZ430-RF2500.

Partition of the information memory, can be locked to separate it from all other segments:

- LOCKA = 1:
  - SegmentA cannot be written or erased;
  - All information memory is protected from erasure during a mass erase or production programming.
- LOCKA = 0:
  - SegmentA can be erased and written;
  - All information memory is erased during a mass erase or production programming.
Flash memory write/erase modes

- Default mode: read mode (memory operates like ROM):
  - Flash memory is not being erased or written;
  - Flash timing generator off;
  - Voltage generator off.

- The flash memory write/erase modes are selected with the BLKWRT, WRT, GMERAS, MERAS, and ERASE bits;

- To stop any write or erase operation before its normal completion, set the EMEX bit. When EMEX = 1:
  - All flash operations cease;
  - The flash returns to read mode;
  - All bits in the FCTL1 register are reset.
Erase modes (1/2)

- **Erase modes:**
  - Initiated from within flash memory:
    - All timing is controlled by the flash controller;
    - CPU is held during the erase cycle (dummy write);
    - CPU resumes code execution after the erase cycle finishes.
  
  - Initiated from RAM:
    - CPU is not held and can execute code from RAM;
    - CPU can access any flash address again when BUSY = 0 (end of the erase cycle).
## Erase modes:

<table>
<thead>
<tr>
<th>Bits</th>
<th>Mode description</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>Segment erase</td>
</tr>
<tr>
<td>0</td>
<td>Mass erase (main memory segments- selected array)</td>
</tr>
<tr>
<td>0</td>
<td>Erase all flash memory (main and information segments – selected array)</td>
</tr>
<tr>
<td>1</td>
<td>Global mass erase (all main memory segments – both arrays)</td>
</tr>
<tr>
<td>1</td>
<td>Erase main memory and information segments- both arrays</td>
</tr>
</tbody>
</table>

1 This bit is only present in the MSP430FG461x devices
2 This bit is only present in the MSP430F2xxx devices
Erase mode procedure (1/3)

- **Segment Erase:**
  - Check BUSY = 0 (FCTL3 register);
  - LOCK = 0 (FCTL3 register);
  - ERASE = 1 (FCTL1 register);
  - Perform a dummy write to the segment to be erased (Any write, clear or logical operation);
  - A segment erase requires approximately 5000 cycles of the timing generator (during this period BUSY = 1);
  - Wait for BUSY = 0 (FCTL3 register).
  - LOCK = 1 (FCTL3 register) to prevent accidental writes.
Mass Erase (all main memory segments):
- Similar to Segment erase;
- Requires setting MERAS bit instead of ERASE bit in the FCTL1 register.

All Erase (all segments):
- Requires setting (GMERAS), MERAS and ERASE bits in the FCTL1 register.
Segment and mass erase modes procedure:

- **Segment and mass erase modes procedure:**
  - **from within flash memory**
  - **from within RAM**

```
1. Disable watchdog
2. Setup flash controller and erase mode
3. Dummy write
4. Set LOCK=1, re-enable watchdog
```
A byte/word write cycle can be initiated from within flash memory or from RAM;

A block write cycle cannot be initiated from within flash memory (only from RAM);

The block write can be used to accelerate the flash write process (twice as fast as byte/word mode), when many sequential bytes or words need to be programmed.

<table>
<thead>
<tr>
<th>Bits</th>
<th>Mode description</th>
<th>Mode description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLKWR</td>
<td>WRT</td>
<td>MSP430FG461x and MSP430F2xxx</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>Byte/word write</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>Block write</td>
</tr>
</tbody>
</table>
Write modes procedure (1/4)

- **Byte/word write:**
  - Check BUSY = 0 (FCTL3 register);
  - LOCK = 0 (FCTL3 register);
  - WRT = 1 (FCTL1 register);
  - Write the byte or word (element) to the appropriate address (starts the timing generator);
  - To write an element requires 33 cycles of the timing generator (during this period BUSY = 1);
  - Wait for BUSY = 0 (FCTL3 register);
  - LOCK = 1 (FCTL3 register) to prevent accidental writes.
Write modes procedure (2/4)

- Byte/word write:
  - From within flash memory:
    1. Disable watchdog
    2. Setup flash controller and set WRT=1
    3. Write byte or word
    4. Set WRT=0, LOCK=1, re-enable watchdog
  - From within RAM:
    1. Disable watchdog
    2. Setup flash controller and set WRT=1
    3. Write byte or word
    4. Check BUSY
    5. If BUSY = 1, repeat setup and write
Write a block (successive write of 64 bytes in a block):

- Check BUSY = 0 (FCTL3 register);
- LOCK = 0 (FCTL3 register);
- WRT = 1 and BLKWRT = 1 (FCTL1 register);
- Write the element in the block to the appropriate address (starts the timing generator);
- Loop until WAIT = 1 (FCTL3 register);
- Repeat write next element until all have been written;
- Set WRT = 0 and BLKWRT = 0 (FCTL1 register);
- A block write requires 20 cycles Timing Generator/element, + overhead: 15 more cycles (during this period BUSY = 1);
- Wait for BUSY = 0 (FCTL3 register);
- LOCK = 1 (FCTL3 register) to prevent accidental writes.
- **Block (64-byte blocks) write:**

  1. Disable watchdog
  2. Setup flash controller
  3. Set BLKWRT=WRT=1
  4. Write byte or word
  5. If WAIT=0?
     - Yes: Continue
     - No: Block Border
  6. Set BLKWRT=0
  7. If BUSY=1?
     - Yes: Another Block
     - No: Set WRT=0, LOCK=1, re-enable WDT
When BUSY = 1, any write or any erase operation initiated from RAM or from flash memory triggers the following conditions:

<table>
<thead>
<tr>
<th>Flash operation</th>
<th>Flash access</th>
<th>Wait</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Any erase</td>
<td>Read</td>
<td>0</td>
<td>ACCVIFG = 0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Value read: 03FFFh</td>
</tr>
<tr>
<td>Any</td>
<td>Write</td>
<td>0</td>
<td>ACCVIFG = 0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Write ignored</td>
</tr>
<tr>
<td>Byte/word write</td>
<td>Instruction fetch</td>
<td>0</td>
<td>ACCVIFG = 0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>CPU fetch: 03FFFh</td>
</tr>
</tbody>
</table>

| Block write     | Any          | 0    | ACCVIFG = 1 |
|                 |              |      | LOCK = 1 |
| Read            | 1            | ACCVIFG = 0 |
|                 |              |      | Value read: 03FFFh |
| Write           | 1            | ACCVIFG = 0 |
|                 |              |      | Flash written |
| Instruction fetch | 1 | ACCVIFG = 1 |
|                 |              |      | LOCK = 1 |
## FCTL1, Flash Memory Control Register (MSP430FG4618)

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>BLKWRT</td>
</tr>
<tr>
<td>6</td>
<td>WRT</td>
</tr>
<tr>
<td>4</td>
<td>EEIX(1)</td>
</tr>
<tr>
<td>3</td>
<td>EEI(1)</td>
</tr>
<tr>
<td></td>
<td>GMERAS(2)</td>
</tr>
<tr>
<td></td>
<td>MERAS</td>
</tr>
<tr>
<td>2</td>
<td>ERASE</td>
</tr>
<tr>
<td>1</td>
<td>ERASE</td>
</tr>
</tbody>
</table>

**Register Details**

- **FCTLx password**
  - Read: FRKEY = 096h
  - Write (must be): FWKEY = 0A5h

- **(1) MSP430F2xx(x) family devices. Not present on MSP430F2013.**
- **(2) MSP430FG461x devices.**
### FCTL2, Flash Memory Control Register (MSP430FG4618)

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>15-14</td>
<td>(FCTLx password)</td>
</tr>
<tr>
<td>13</td>
<td>Read: FRKEY = 096h</td>
</tr>
<tr>
<td>12</td>
<td>Write (must be): FWKEY = 0A5h</td>
</tr>
<tr>
<td>11-4</td>
<td>FSSELx Flash controller clock source:</td>
</tr>
<tr>
<td>3-0</td>
<td>FNx Flash controller clock divider</td>
</tr>
<tr>
<td>7-6</td>
<td>FSSEL1 FSSEL0 = 00 ⇒ ACLK</td>
</tr>
<tr>
<td></td>
<td>FSSEL1 FSSEL0 = 01 ⇒ MCLK</td>
</tr>
<tr>
<td></td>
<td>FSSEL1 FSSEL0 = 10 ⇒ SMCLK</td>
</tr>
<tr>
<td></td>
<td>FSSEL1 FSSEL0 = 11 ⇒ SMCLK</td>
</tr>
<tr>
<td>5-0</td>
<td>FNx = 00h ⇒ /1</td>
</tr>
<tr>
<td></td>
<td>...</td>
</tr>
<tr>
<td>0</td>
<td>FNx = 03Fh ⇒ /64</td>
</tr>
</tbody>
</table>

---

Copyright 2009 Texas Instruments
All Rights Reserved
www.msp430.ubi.pt
### FCTL3, Flash Memory Control Register (MSP430FG4618)

(FCTLx password)  
Read: FRKEY = 096h  
Write (must be): FWKEY = 0A5h

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>FAIL</td>
</tr>
<tr>
<td>6</td>
<td>LOCKA(1)</td>
</tr>
<tr>
<td>5</td>
<td>EMEX</td>
</tr>
<tr>
<td>4</td>
<td>LOCK</td>
</tr>
<tr>
<td>3</td>
<td>WAIT</td>
</tr>
<tr>
<td>2</td>
<td>ACCVIFG</td>
</tr>
<tr>
<td>1</td>
<td>KEYV</td>
</tr>
<tr>
<td>0</td>
<td>BUSY</td>
</tr>
</tbody>
</table>

#### Bit Descriptions

- **FAIL**: Operation failure of the clock source, $f_{(FTG)}$, or a flash operation is aborted from an interrupt when EEIEX = 1 when FAIL = 1.
- **LOCKA(1)**: Segment A locked and all information memory is protected from erasure during a mass erase when LOCKA = 1.
- **EMEX**: Emergency exit when EMEX = 1.
- **LOCK**: Locks the flash memory for writing or erasing when LOCK = 1.
- **WAIT**: WAIT = 0 $\Rightarrow$ while flash memory is being written to  
  WAIT = 1 $\Rightarrow$ when flash memory is ready for the next byte/word write.
- **ACCVIFG**: Access violation interrupt flag ACCVIFG = 1 when interrupt is pending.
- **KEYV**: Flash security key violation KEYV = 1 when FCTLx password was written incorrectly (not 0A5h).
- **BUSY**: Flash timing generator is busy when BUSY = 1.
The Tag-Length-Value (TLV) structure is used in selected MSP430x2xx devices to provide device-specific information in the device’s flash memory Segment A, such as calibration data.
The first two bytes of SegmentA (0x10C0 and 0x10C1) hold the checksum of the remainder of the segment (addresses 0x10C2 to 0x10FF).

The first tag is located at address 0x10C2:
- In this example, the TAG_EMPTY tag.

The following byte (0x10C3) holds the length of the structure:
- The length of this TAG_EMPTY structure is 0x16;
- The next tag, TAG_ADC12_1, is found at address 0x10DA;
- The following byte holds the length of the TAG_ADC12_1 structure.
The TLV structure maps the entire address range 0x10C2 to 0x10FF of the SegmentA:

- A program routine looking for tags starting at the SegmentA address 0x10C2 can extract all information even if it is stored at a different (device-specific) absolute address.
- Each device contains a subset of the tags:
  - See the device-specific data sheet for details.

<table>
<thead>
<tr>
<th>Tag</th>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>TAG_EMPTY</td>
<td>Identifies an unused memory area</td>
<td>0xFE</td>
</tr>
<tr>
<td>TAG_DCO_30</td>
<td>Calibration values for the DCO at room temperature and $D_{VC} = 3$ V</td>
<td>0x01</td>
</tr>
<tr>
<td>TAG_ADC12_1</td>
<td>Calibration values for the ADC12 module</td>
<td>0x08</td>
</tr>
</tbody>
</table>

- TAG_ADC12_1 Calibration TLV Structure:
  - Consists of eight words.

<table>
<thead>
<tr>
<th>Label</th>
<th>Description</th>
<th>Offset</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAL_ADC_25T85</td>
<td>$V_{REF2} = 1$, $T_A = 85^\circ C \pm 2K$, 12-bit conversion result</td>
<td>0x0E</td>
</tr>
<tr>
<td>CAL_ADC_25T30</td>
<td>$V_{REF2} = 1$, $T_A = 30^\circ C \pm 2K$, 12-bit conversion result</td>
<td>0x0C</td>
</tr>
<tr>
<td>CAL_ADC_25VREF_FACTOR</td>
<td>$V_{REF2} = 1$, $T_A = 30^\circ C \pm 2K$</td>
<td>0x0A</td>
</tr>
<tr>
<td>CAL_ADC_15T85</td>
<td>$V_{REF2} = 0$, $T_A = 85^\circ C \pm 2K$, 12-bit conversion result</td>
<td>0x08</td>
</tr>
<tr>
<td>CAL_ADC_15T30</td>
<td>$V_{REF2} = 0$, $T_A = 30^\circ C \pm 2K$, 12-bit conversion result</td>
<td>0x06</td>
</tr>
<tr>
<td>CAL_ADC_15VREF_FACTOR</td>
<td>$V_{REF2} = 0$, $T_A = 30^\circ C \pm 2K$</td>
<td>0x04</td>
</tr>
<tr>
<td>CAL_ADC_OFFSET</td>
<td>$V_{REF} = 2.5V$, $T_A = 85^\circ C \pm 2K$, $f_{ADC12CLK} = 5$ MHz</td>
<td>0x02</td>
</tr>
<tr>
<td>CAL_ADC_GAIN_FACTOR</td>
<td>$V_{REF} = 2.5V$, $T_A = 85^\circ C \pm 2K$, $f_{ADC12CLK} = 5$ MHz</td>
<td>0x00</td>
</tr>
</tbody>
</table>
Temperature Sensor Calibration Data:
- The temperature sensor is calibrated using the internal voltage references;
- At VREF2_5 = 0 and 1, the conversion result at 30°C and 85°C is written at the respective SegmentA location;
- See Table for TAG_ADC12_1 Calibration Data.

Integrated Voltage Reference Calibration Data
- The reference voltages (VREF2_5 = 0 and 1) are measured at room temperature;
- The measured value is normalized by 1.5/2.5V before stored into the flash information memory SegmentA:

\[
\text{CAL_ADC}_{-15}\text{VREF\_FACTOR} = \frac{V_{e\text{REF}}}{1.5V} \times 2^{15}
\]
The conversion result is corrected by multiplying it with the CAL_ADC_15VREF_FACTOR (or CAL_ADC_25VREF_FACTOR) and dividing the result by 2^{15}:

$$ADC(\text{corrected}) = ADC(\text{raw}) \times \text{CAL\_ADC\_15VREF\_FACTOR} \times \frac{1}{2^{15}}$$

- **Offset and Gain Calibration Data:**
  - The offset of the ADC12 is determined and stored as a two's-complement number in SegmentA;
  - The offset error correction is done by adding the CAL_ADC_OFFSET to the conversion result:

$$ADC(\text{offset\_corrected}) = ADC(\text{raw}) + \text{CAL\_ADC\_OFFSET}$$
The gain of the ADC12, stored at offset 0x00, is calculated by the following equation:

\[
\text{CAL\_ADC\_GAIN\_FACTOR} = \frac{1}{\text{GAIN}} \times 2^{15}
\]

The conversion result is gain corrected by multiplying it with the \text{CAL\_ADC\_GAIN\_FACTOR} and dividing the result by \(2^{15}\):

\[
\text{ADC(gain_corrected)} = \text{ADC(raw)} \times \text{CAL\_ADC\_GAIN\_FACTOR} \times \frac{1}{2^{15}}
\]

If both gain and offset are corrected, the gain correction is done first.

\[
\text{ADC(gain_corrected)} = \text{ADC(raw)} \times \text{CAL\_ADC\_GAIN\_FACTOR} \times \frac{1}{2^{15}}
\]

\[
\text{ADC(final)} = \text{ADC(gain_corrected)} + \text{CAL\_ADC\_OFFSET}
\]
DCO Calibration TLV Structure:

- The BCS+ registers (BCSCTL1 and DCOCTL) are used.
- The values stored in the flash information memory SegmentA are written to the BCS+ registers.

<table>
<thead>
<tr>
<th>Label</th>
<th>Description</th>
<th>Offset</th>
</tr>
</thead>
<tbody>
<tr>
<td>CALBC1_1MHZ</td>
<td>Value for the BCSCTL1 register for 1 MHz, T_A = 25°C</td>
<td>0x07</td>
</tr>
<tr>
<td>CALDCO_1MHZ</td>
<td>Value for the DCOCTL register for 1 MHz, T_A = 25°C</td>
<td>0x06</td>
</tr>
<tr>
<td>CALBC1_8MHZ</td>
<td>Value for the BCSCTL1 register for 8 MHz, T_A = 25°C</td>
<td>0x05</td>
</tr>
<tr>
<td>CALDCO_8MHZ</td>
<td>Value for the DCOCTL register for 8 MHz, T_A = 25°C</td>
<td>0x04</td>
</tr>
<tr>
<td>CALBC1_12MHZ</td>
<td>Value for the BCSCTL1 register for 12 MHz, T_A = 25°C</td>
<td>0x03</td>
</tr>
<tr>
<td>CALDCO_12MHZ</td>
<td>Value for the DCOCTL register for 12 MHz, T_A = 25°C</td>
<td>0x02</td>
</tr>
<tr>
<td>CALBC1_16MHZ</td>
<td>Value for the BCSCTL1 register for 16 MHz, T_A = 25°C</td>
<td>0x01</td>
</tr>
<tr>
<td>CALDCO_16MHZ</td>
<td>Value for the DCOCTL register for 16 MHz, T_A = 25°C</td>
<td>0x00</td>
</tr>
</tbody>
</table>
The 64-byte SegmentA contains a 2-byte checksum of the data stored at 0x10C2 up to 0x10FF at addresses 0x10C0 and 0x10C1.

- The checksum is a bit-wise XOR of 31 words stored in the twos-complement data format.