Technical notes on using Analog Devices DSPs, processors and development tools
Visit our Web resources http://www.analog.com/ee-notes and http://www.analog.com/processors or e-mail processor.support@analog.com or processor.tools.support@analog.com for technical support.

Interfacing Atmel® Fingerprint Sensor AT77C104B with Blackfin® Processors

Contributed by Jayanti Addepalli and Aseem Vasudev *Rev 1 – August 9, 2007*

Introduction

Object recognition is one of the most interesting human abilities. It involves perception and the ability to associate the resulting information with one or a combination of its memory contents. This EE-Note emulates this human ability with regard to perception. It demonstrates the implementation of the interface between an ADSP-BF533 Blackfin® processor and an Atmel® AT77C104B FingerChip® thermal fingerprint sensor.

The need for effective security is evident in today's world. Specific security concerns include protecting computer systems, PDAs, mobile phones, smart phones, Internet appliances, and similar devices from unauthorized access or use. Biometrics seeks to identify individuals uniquely by measuring certain distinctive physiological and behavioral characteristics, referred to as biometric identifiers or biometrics. A sample is obtained from certain measurements in a standard data format, which is compared with a template, obtained using the same standard metrics. A close match between the sample and template confirms the identity of an individual.

Fingerprints are distinct, permanent, and one of the most widely accepted Biometric identifiers. A fingerprint is formed of composite curve segments. The light areas of fingerprints are called ridges, and the dark areas are called valleys. Minutiae, the local discontinuities in the ridge flow pattern, can be used as discriminating features since they are unique and permanent.

A fingerprint sensor "reads" the finger surface and converts the analog reading into digital form through an analog-to-digital converter (ADC); an interface module is responsible for communicating with an external processor. Live fingerprint scanners can be broadly classified as optical, solid-state, and ultrasound sensors. Solid-state sensors include capacitive, thermal, piezo-electric, and electric field-based sensors. The Atmel AT77C104B FingerChip IC is a thermal fingerprint sensor for fingerprint image capture. It combines detection and data conversion circuitry in a single rectangular CMOS die. It is a linear sensor that captures fingerprint images by sweeping the finger over a sensing area.

The following sections of this document detail the sensor command format and demonstrate with example code how this sensor can be interfaced seamlessly with a Blackfin processor's SPI port. The example code in the associated `.ZIP` file implements various commands used to configure and operate the fingerprint sensor.

Atmel FingerChip Technology – AT77C104B

Atmel's AT77C104B FingerChip IC for fingerprint image capture works on thermal sensing technology. A pyro-electric material generates current, based on temperature differentials. The fingerprint ridges, being in contact with the sensor surface, produce a different temperature differential than valleys, which are away from the sensor surface. This temperature differential produces an image when contact occurs, but this image soon disappears because a thermal equilibrium is reached, stabilizing the pixel temperature. Hence, a sweeping method is necessary to acquire a stable fingerprint image. The sensor captures the image of a fingerprint as the finger is swept vertically over the sensor window, as shown in Figure 1. It requires no external heat, light, or radio source.

Figure 1. Finger being swept over a sensor

The IC has an embedded temperature stabilization unit that identifies the difference in temperature between the finger and the sensor. When this difference is increased, the images have more contrast.

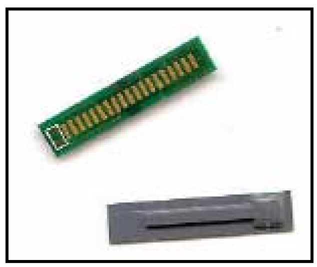

Figure 2. AT77C104B FingerChip IC

Figure 2 shows the actual size of the IC. The sensor is available with a holder, as shown in Figure 3, to integrate it easily with the PCB.

Figure 3. AT77C104B sensor IC with holder and elastomer connector

Never solder this sensor. The thermal sensor uses an elastomer to establish electrical connection between the sensor leads and PCB pads. Soldering this sensor may cause permanent damage.

The elastomer connectors have the ability to make closer connections. They have alternating layers of conductive (carbon/ silver filled/ conductive silicone rubber) and non-conductive (non-conductive silicone rubber) material.

A PCB must be designed so that the sensor-holder fits exactly into the holes. The conductive connections of the elastomer must be in contact with the copper pads on the PCB. The details for the mechanical design of the PCB are given in the AT77C104B sensor data sheet[2]. When designed, the PCB would look like Figure 4. The sensor with its holder should be press-fitted as shown in the arrangement of Figure 5.

Figure 4. PCB designed for the sensor

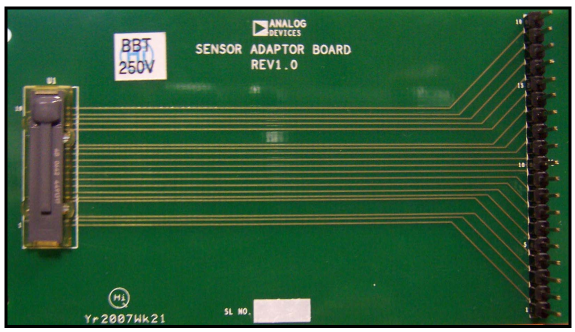

Figure 5. Sensor on PCB

The benefits of FingerChip technology derive from its thermal sensing technique, its frame-sweeping method of image capture, and the integration of the sensor and data conversion circuitry as a single IC. The finger sweep technology ensures that the sensor surface does not become dirty; a periodic cleaning is not required. Latent fingerprints do not remain on the sensor once the finger has been removed, unlike in touch-based sensors.

AT77C104B Features, Command Format, and Transfer Protocol

The FingerChip sensor contains an array of 8 rows by 232 columns, giving 1856 temperature-sensitive pixels. An additional dummy column is used for calibration and frame identification. The dummy column at the beginning of the pixel array is added to represent the start of a frame.

AT77C104B Features

The main parameters that characterize a fingerprint sensor include resolution, area, number of pixels, dynamic range, and geometrical accuracy. Brief descriptions of these and other parameters follow.

- Resolution: Refers to the number of dots or pixels per inch (dpi). Minutiae play a primary role in fingerprint matching, since most algorithms rely on the co-incidence of minutiae to declare whether the two fingerprint impressions are of the same finger. Higher resolution results in better resolve between ridges and valleys, hence, finer isolation of minutiae points. The AT77C104B sensor has a resolution of 500 dpi.

- Area: The size of the rectangular area sensed by a fingerprint sensor is a fundamental parameter. In general, a larger area results in a more distinctive fingerprint. But in the case of the FingerChip sensor, the finger-sweep technology enables the sensor to provide the same features with a smaller area of 0.4 x 11.6 mm. This is an advantage because a larger die costs more, due to fewer dies per wafer and lower yield.
- Dynamic range or depth: Denotes the number of bits used to encode the intensity value of each pixel. The AT77C104B sensor uses four bits to encode each pixel. This results in 16 grayscale levels.
- Number of pixels: The number of pixels in a fingerprint image can be derived by the resolution and fingerprint area of 8 x 232 (1856) pixels per frame.
- Serial Peripheral Interface (SPI): Two modes: fast SPI mode at 16 Mbps (max) for imaging, and slow SPI mode at 200 Kbps (max) for navigation and control.
- Operating voltage from 2.3 V to 3.6 V and operating temperature range of -40°C to +85°C.
- On-chip 8-bit analog-to-digital converter
- Finger sweeping speed from 2 to 20 cm/second, low power consumption, and small form factor packaging.
- High protective coating (>4 million sweeps): High protection from electrostatic discharge and compliance with the RoHS Directive.

Figure 6 shows the block diagram of the fingerprint sensor. The internal circuit consists of an array or frame of 8 x 232 pixels and one dummy column. It also consists of an analog-to-digital converter, an on-chip oscillator, control and status registers, and navigation and click units. It has separate interfaces for slow and fast modes of operation.

The sensor has a provision for heating. In order to limit excessive current consumption by the use of the temperature stabilization function, a watchdog timer has been implanted in the sensor. The local oscillator stops heating the module after a specified length of time. The oscillator should not be stopped as long as the watchdog timer is active; otherwise, the clock stops automatically. The level of power consumption is programmable. Two pre-programmed values are set to 50 mW or 100 mW.

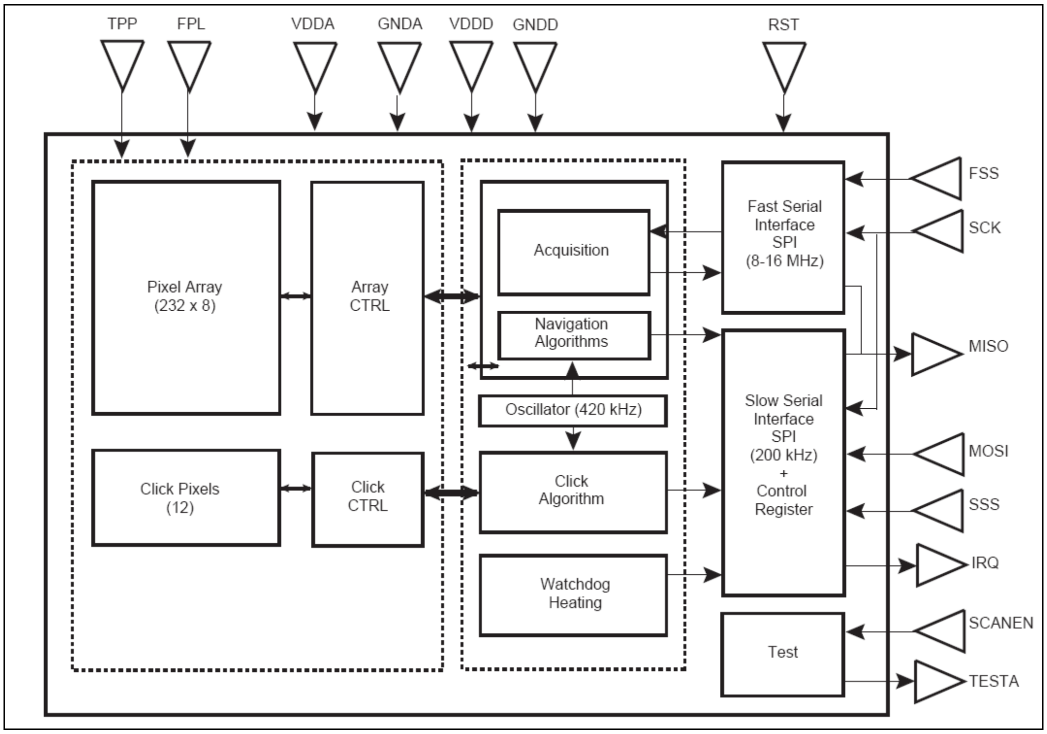

Figure 6. Block diagram of the fingerprint sensor

Command Format and Transfer Protocol

This section describes the sensor's communication protocol to transfer commands and data. The AT77C104B sensor uses the Serial Peripheral Interface (SPI) to communicate with a host device (digital signal processor or microcontroller). It supports two SPI modes (fast mode and slow mode).

Slow SPI mode is used to program, control, and configure the sensor. It can run the SPI interface at a maximum speed of 200 kHz. Fast mode is used for data acquisition from the sensor. It can operate the serial interface up to 16 MHz (the minimum speed of operation is 8 MHz). The `/SSS` signal enables the slow SPI mode, and `/FSS` enables fast SPI mode.

The AT77C104B sensor has a command interface and accepts commands over the slow SPI. It has a group of registers that are used to configure various parameters, enable modes such as click mode, navigation mode, and data acquisition mode. The `STATUS` register is read to identify the state of sensor. It is read whenever the sensor sends a request in form of an interrupt (`/IRQ`) signal, in order to identify the cause of the interrupt. Table 1 shows a list of all internal registers of the sensor.

Register	Address (b3 down to b0)	Read/Write
STATUS	0000	Read
MODECTRL	0001	Read/Write
ENCTRL	0010	Read/Write
HEATCTRL	0011	Read/Write
NAVCTRL	0100	Read/Write
CLICKCTRL	0101	Read/Write
MOVCTRL	0110	Read/Write
	0111	Reserved
NAVIGATION	1000	Read
NAVIGATION	1001	Reserved
NAVIGATION	1010	Reserved
PIXELCLICK	1011	Reserved
PIXELCLICK	1100	Reserved
PIXELCLICK	1101	Reserved
	1110	Reserved

Table 1. Internal registers of the AT77C104B sensor

Internal registers are identified by their unique 4-bit address. The host uses a command format to indicate to the sensor if it wants to read or write to a register. The register's address forms a part of this command. Every transfer to and from the sensor is an 8-bit data transfer. Figure 7 shows the command format.

b7	b6	b5	b4	b3	b2	b1	b0
1	Read(1)/ Write(0)	Address (b3)	Address (b2)	Address (b1)	Address (b0)	X	X

Figure 7. Sensor command format

The transfer starts with the most significant bit (MSB) being sent or received first. An MSB (b7) of '1'indicates that the word is a command word; '0' refers to a data word. Bit 'b6' indicates whether the command is a read or a write command. If it is '1', the command is a read command; if it is '0', it refers to a write command. The bits from 'b5' to 'b2' indicate the address of sensor's internal register to be read/written into. Bits 'b1' and 'b0' are don't care bits. Figure 8 shows the data format.

b7	b6	b5	b4	b3	b2	b1	b0
0	Data (b6)	Data (b5)	Data (b4)	Data (b3)	Data (b2)	Data (b1)	Data (b0)

Figure 8. Sensor data format

The data written to the sensor register is always 7 bits as the MSB (b7) is used to indicate that it is a data word. Data read back from the sensor registers is also 7 bits, other than for the `STATUS` register, in which 8 bits are read and the navigation register, in which 24 bits are read.

Modes of Operation

The sensor implements six modes of operation as discussed below:

- **Sleep mode:** A very low consumption mode controlled by the reset pin (`RST`). In this mode, internal clocks are disabled and the registers are initialized.
- **Stand-by mode:** A low consumption mode that waits for an action from the host. The slow serial port interface (SSPI) and control blocks are activated. The oscillator may remain active in this mode.
- **Click mode:** Waits for a finger on the sensor. The SSPI and control blocks are activated. The local oscillator, click array, and click block are activated.
- **Navigation mode:** Calculates the finger's x and y movements across the sensor. The SSPI and control blocks are still activated. The local oscillator, navigation array, and navigation block are also activated.
- **Acquisition mode:** Slices are sent to the host for fingerprint reconstruction and identification. The SSPI and control blocks are still activated. The fast serial port interface block (FSPI) and the acquisition array are activated, as well as the local oscillator when a watchdog timer is required.
- **Test:** This mode is reserved for factory testing.

In the final application, three main modes are used:

- **Stand-by:** Low consumption mode
- **Pointing**: Equivalent to click and navigation modes
- **Acquisition:** Fingerprint image capture

Serial Peripheral Interface (SPI) of the Blackfin Processor

The Blackfin processor's SPI is a four-wire interface consisting of two data pins (`MOSI` and `MISO`), a device select pin (`/SPISS`), and a gated clock pin (`SCK`). The SPI is a full-duplex synchronous serial interface, supporting master modes, slave modes, and multi-master environments. The SPI-compatible peripheral implementation also supports programmable baud rate and clock phase/polarities. Figure 9 shows a block diagram of a Blackfin SPI.

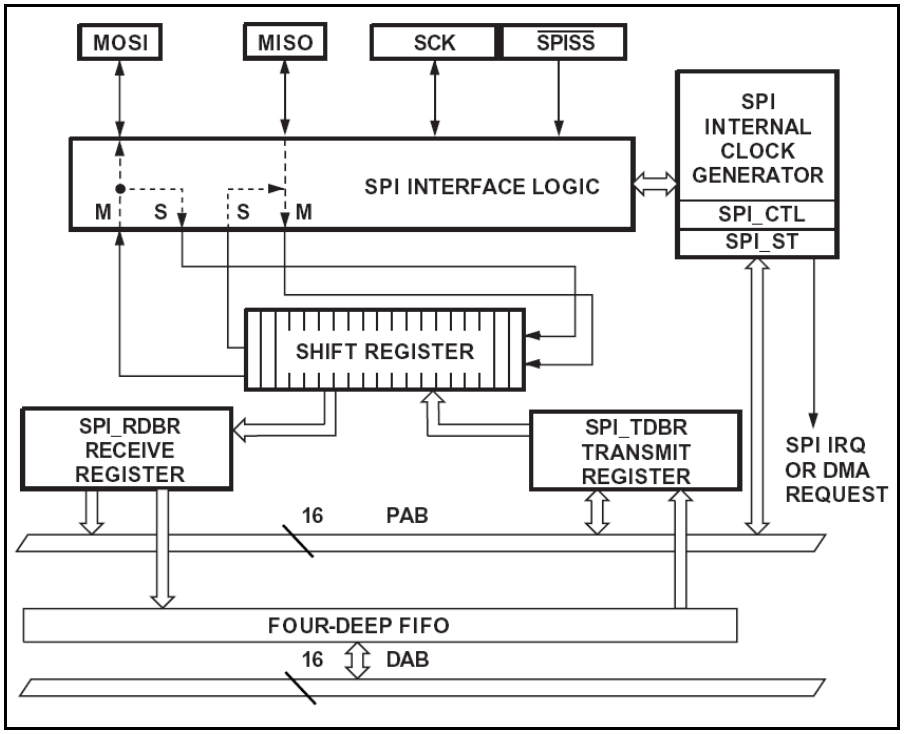

Figure 9. Block diagram of Blackfin processor's SPI port

The interface is essentially a shift register that serially transmits and receives data bits, one bit at a time at the `SCK` rate, to and from other SPI devices. SPI data is transmitted and received at the same time through the use of a shift register. The `SCK` synchronizes the shifting and sampling of the data on the two serial data pins.

SPI can be configured as a master SPI port (generates `SCK` and `/SPISS` signals) or can be configured as an SPI slave port (receives `SCK` and slave select signals externally). When the SPI port is configured as master, it drives data on the `MOSI` pin and receives data on the `MISO` pin. It drives the slave select signals for SPI slave devices and provides the serial bit clock (`SCK`). The Blackfin processor's SPI supports four functional modes by using combinations provided by the clock polarity (`CPOL`) and clock phase (`CPHA`) bits. For detailed information on the Blackfin SPI port, refer to the *ADSP-BF533 Blackfin Processor Hardware Reference Manual* [1].

Hardware Interface

The hardware interface between the ADSP-BF533 processor's SPI port and the AT77C104B is seamless and does not require any external glue logic. The slave select signals of the sensor (`/SSS` and `/FSS`) are driven through programmable flag pins `PF1` and `PF2`, respectively. The sensor generates interrupts through the `/IRQ` pin, which is connected to `PF4`, configured as input. The reset to the IC (`RST`) is driven through the `PF3` signal. 10K pull-up resisters are required on the `/IRQ`, `MISO`, `MOSI`, `SCK`, `/SSS` and `/FSS`, signals. A pull-down resistor is recommended on the `RST` signal. Table 2 describes the pin functions of the AT77C104B sensor. The arrangement of pins is as shown in Figure 10.

Pin Number	Name	Type	Description
1			Not Connected – NC
2			Not Connected – NC
3			Not Connected – NC
4			Not Connected – NC
5	GNDD	G	Digital Ground Supply
6	GNDA	G	Analog Ground Supply – Connect to GNDD
7	VDDD	P	Digital Power Supply
8	VDDA	P	Analog Power Supply – Connect to VDD
9	SCK	I	Serial Port Interface SPI bit clock
10	TESTA	IO	Reserved for analog test – Not Connected NC
11	MOSI	I	Master out slave in data – reads data on this pin
12	TPP	P	Temperature stabilization power
13	MISO	O	Master in slave out data – sends data on this pin
14	SCANEN	I	Reserved for scan test in factory, must be grounded
15	/SSS	I	Slow SPI slave select – active low
16	/IRQ	O	Interrupt line to host – active low
17	/FSS	I	Fast SPI slave select – active low
18	RST	I	Reset and sleep mode control – active high
19	FPL	I	Front plan – Must be grounded

Table 2. Pin description for chip-on-board package AT77C104B - CH08YV

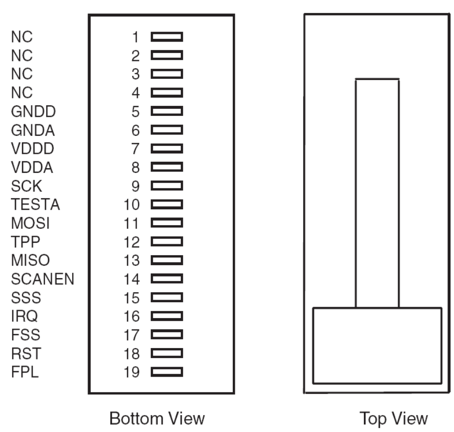

Figure 10. Pin description

`/SSS` and `/FSS` must never be driven low at the same time. When both `/SSS` and `/FSS` are driven low, the sensor chip switches to scan test mode.

The schematic diagram for this interface is shown in Figure 11. 10µF capacitors are recommended between the VDD and ground pins. Pull-up resistors on `/SSS` and `/FSS` ensure that the SPI ports of the sensor are disabled during system power-up, when the ADSP-BF533 processor does not drive these signals. The ADSP-BF533 processor uses two general-purpose flag pins to drive `/SSS` and `/FSS` of the sensor. These flags should never be configured as output flags simultaneously, because the Blackfin processor drives them logic low by default when configured as output flags. Therefore, this would drive `/SSS` and `/FSS` active at the same time switching the sensor chip to scan test mode. The software should configure one of these flags as an output and drive logic high on it before configuring the other flag as output. The application software uses `PF3` of the Blackfin processor to drive the `RST` (reset) of the sensor. The sensor reset is an active high signal, so a 10K pull-down resistor is used on this line. A 10K pull-up resistor has been used on the `SCK` signal.

A 10K pull-up resistor must be used on the `MISO` line. The EZ-KIT Lite® evaluation board has a pull-down resistor on the `MISO` pin. This creates a voltage-divider circuit when the `MISO` pin is pulled high. Therefore, this resistor must be removed.

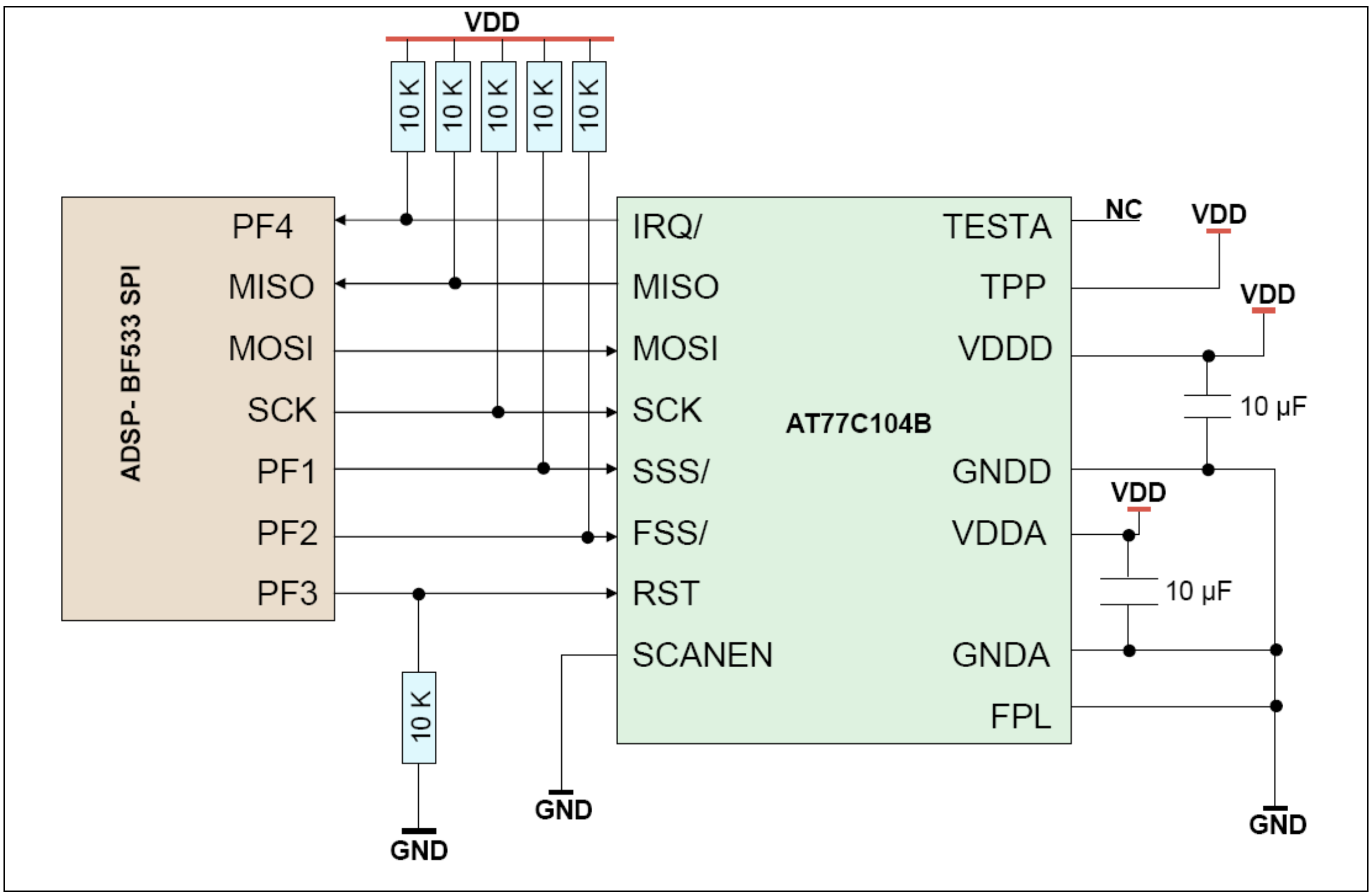

Figure 11. Schematic diagram of interface between ADSP-BF533 processor and AT77104B FingerChip sensor

Sensor Chip Initialization

Before performing any operations, the thermal sensor must be initialized. This sequence normally happens after power-up and before performing any read or write operations to the sensor. The sensor reset (`RST`) must be driven active (logic high) for a minimum duration of 10 µsec. SPI chip enables (`/SSS` and `/FSS`) of the sensor must be driven in-active (logic high) during this chip initialization sequence. Either of the sensor's SPI ports can be enabled only after a minimum period of "T_{RSTSU}" from the falling edge of the `RST` signal. Figure 12 shows the sensor chip initialization sequence. The `RST` signal is also used to switch the sensor to sleep mode, which is a very low power consumption mode.

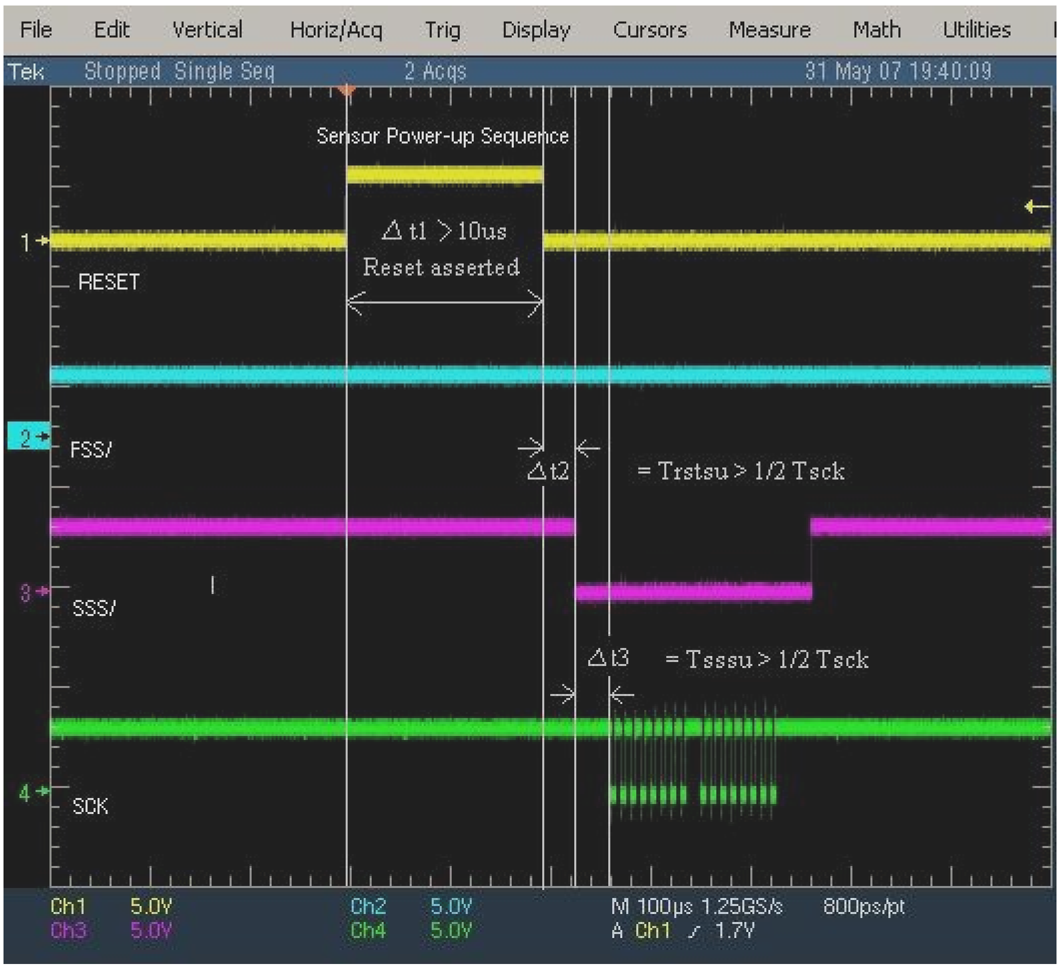

Figure 12. Sensor chip initialization sequence

Blackfin SPI Configuration and Sensor SPI Timing Specifications

The Blackfin SPI is always the master, driving `SCK` and slave chip-select signals. It is configured in core mode while using the slow SPI of sensor. For data acquisition, it is configured in DMA mode.

In core mode, the `TIMOD` bits are programmed such that the SPI transfer is initiated on the read of the SPI receive (`SPI_RDBR`) register. An SPI interrupt is generated when the receive SPI buffer is full. In the case of DMA transfers during data acquisition phase, the `TIMOD` bits are programmed to initiate SPI transfer on a read of receive the FIFO and an SPI interrupt is generated, if programmed, when the receive count of that DMA channel expires.

The SPI signal timings specified in the sensor data sheet are critical and should never be violated. The combination of the `CPHA` and `CPOL` bits in the SPI control registers should be such that the `SCK` level remains high during the inactive phase and the data is driven out on `MOSI` on the falling edges of `SCK`. This requirement translates to the combination: `CPHA` = 1 and `CPOL` = 1. The sensor supports only this mode of SPI. It drives data on the falling `SCK` edges and reads incoming data on the rising edges of `SCK`. The delay between the falling edge of `/SS` (`/SSS` or `/FSS`) and the first falling edge of `SCK` (T_{SSSU}) must be a minimum of ½ T_{SCK}. Also, the delay between the last (rising) edge of `SCK` and the rising edge of `/SS` (T_{SSHD}) must be a minimum of ½ T_{SCK}. These timing requirements are met using delay functions in software.

SPI Behavior with Hazardous Access

The control register block of the sensor uses an internal finite state machine that can only be initialized by the `RST` pin (asynchronous reset). When an SPI access does not use eight clock pulses, the internal finite state machine is desynchronized. The only way to resynchronize it is by resetting the sensor with the `RST` pin. No requester modification is recorded when a write access is made on a read-only register. Reliable initialization of read-only registers is not guaranteed when the slow SPI's maximum clock frequency is not respected.

Application Software

The application code supplied with this EE-Note performs tasks such as controlling the sensor, acquiring fingerprint data from the sensor, and re-arranging this raw data so as to display the received fingerprint image using the Image Viewer plug-in of the VisualDSP++® development tools. Blackfin processors use two interrupts (one for `/IRQ` drive by sensor and the other for SPI) and therefore have two interrupt routines running. The `/IRQ` interrupt indicates a pending request from the thermal sensor.

SPI interrupt occurs due to either core SPI transfer or a DMA SPI transfer. Core SPI interrupt is used to transfer commands, and data related to these commands to and from the sensor. DMA interrupt indicates completion of the data acquisition phase.

The code flow is briefly described next:

- Program Blackfin PLL – VCO, core clock and system clock
- Initialize external SDRAM – to store fingerprint data received from the AT77C104B
- FingerChip sensor initialization – power-up sequence
- Configure required sensor registers for various parameters using slow SPI mode
- Switch sensor to click-detect mode and enable click interrupt
- After detecting a click, switch sensor to navigation mode and enable navigation interrupt
- After detecting navigation, switch sensor to acquisition mode
- Configure the heat control and watchdog timer settings

- Check for any read error interrupt. If found, quit and re-start.
- Acquire fingerprint data from sensor using its fast SPI mode
- Wait for the DMA completion interrupt
- Check for read error interrupt after DMA completion interrupt. If found, quit and re-start.
- Re-arrange process received data
- Sequence back to the step where the sensor is switched to click-detect mode

When the sensor detects a click, it generates an interrupt (`/IRQ`) to the host (Blackfin processor). The Blackfin processor receives this interrupt through the `PF4` pin, which is configured as input, to generate an interrupt on a falling edge. The application then reads the `STATUS` register to determine the sensor request. A read of the `STATUS` register clears the interrupt (`/IRQ` is driven logic high again). The `STATUS` register is 8 bits, and its value indicates the event causing an interrupt. The same process repeats for other interrupts like the navigation interrupt and the read error interrupt. The complete flow for the application is briefly shown in the flow chart in Figure 13.

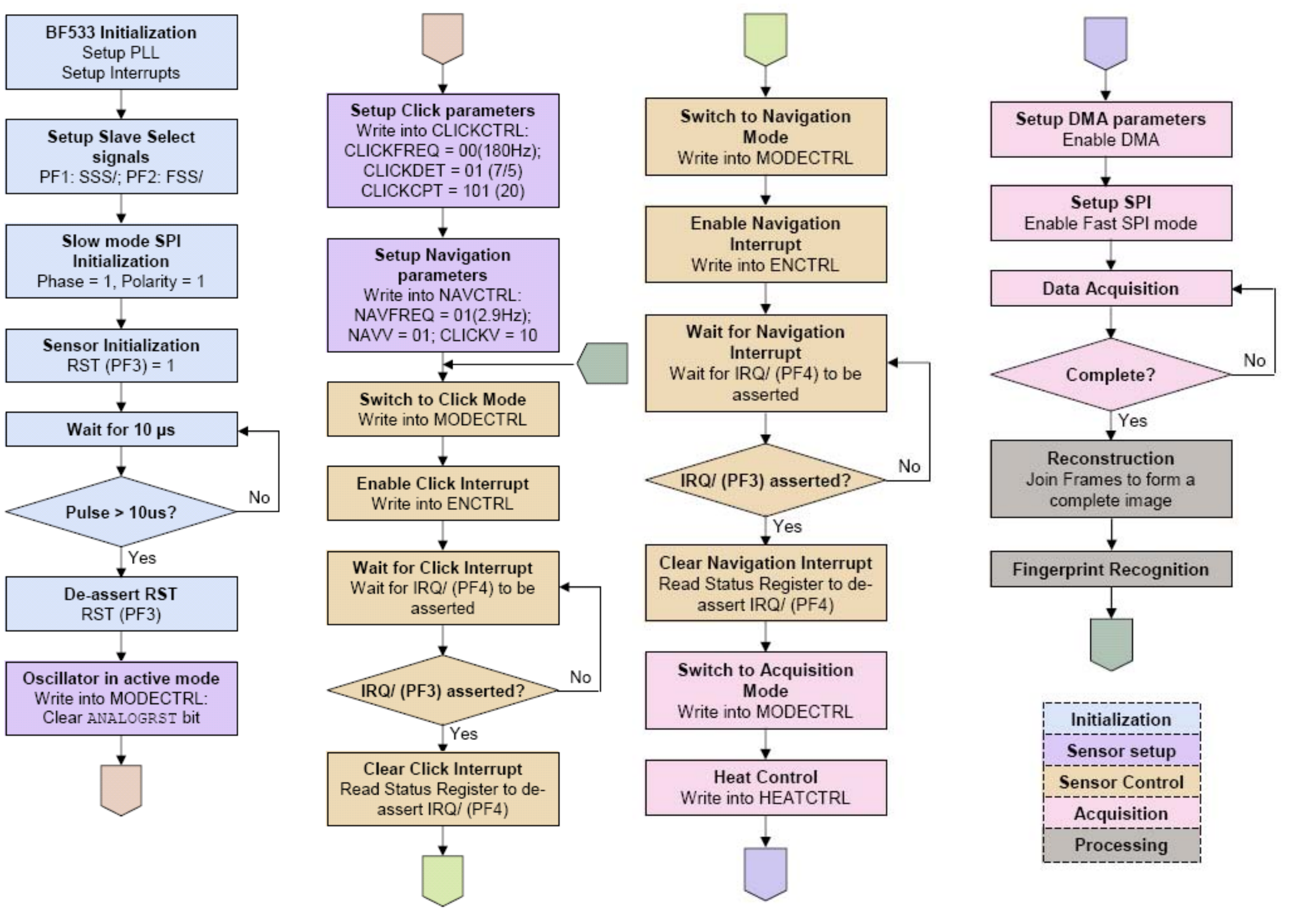

Figure 13. Application flow chart

AT77C104B Configuration and Mode Switching

The command word and data to be written to sensor registers are passed to the `WriteSensorRegsiter()` and `ReadSensorRegister()` functions using a global command array. The SPI in core mode is programmed to initiate data transfer on a read of the receive SPI buffer. The first word (command and address) to be sent to the sensor is written to transmit the SPI buffer, and a dummy read of receive buffer is performed. The dummy read initiates the SPI transfer, and a receive buffer full condition generates an SPI interrupt. The next value from the global command array is written to the SPI transmit buffer in the interrupt service routine. The received data from the receive SPI buffer is read into the global data array. This initiates the second SPI transfer. A variable, `SPITransferCount`, is used to track the number of SPI transfers. This count is incremented in the core SPI ISR. When it reaches the expected count (N), the Blackfin SPI is disabled and the slave select flag (`/SSS`) is de-asserted after the specified delay of 1/2 T_{SCK} cycle.

Figure 14 through Figure 18 show various sensor operations such as register write, register read, click interrupt sequence, navigation interrupt sequence, and data acquisition.

Figure 14. Write MODECTRL register

Slow SPI chip select (`/SSS`) is driven logic low to select the slow SPI mode and to perform a write to the `MODECTRL` register. A sufficient delay is introduced between the falling edge of `/SSS` and the first (falling) edge of SCK. A minimum delay of "T_{SSSU} = ½ T_{SCK}" is required. A command word consisting of a 4-bit MODECTRL register address is sent to MOSI. The actual data to be written to MODECTRL register is sent out as the second word. In this case, the SPI interface is operating at about 100 kHz.

Figure 15 shows the read operation of the MODECTRL register. The sensor drives data of the MODECTRL register on the MISO pin during the data word time slot.

Figure 15. Read MODECTRL register

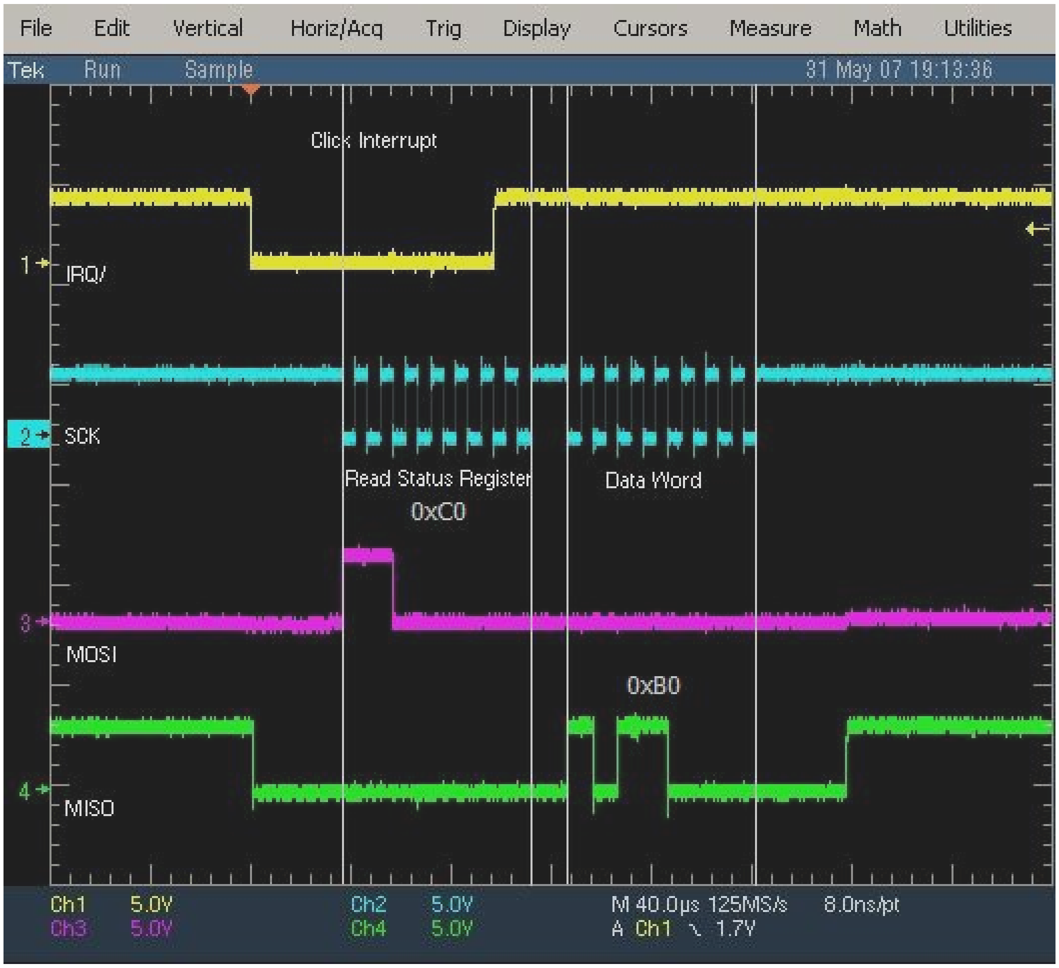

Figure 16. Click interrupt and read status register

A click is detected by the sensor when it is touched, and a click interrupt is generated. The Blackfin processor reads the `STATUS` register to detect what event caused an interrupt. The read operation of the `STATUS` register clears the interrupt, and the `/IRQ` signal is driven logic high by the sensor. The `STATUS` register data "data word" is driven by the sensor on `MISO`. Its value indicates that a click interrupt has been detected.

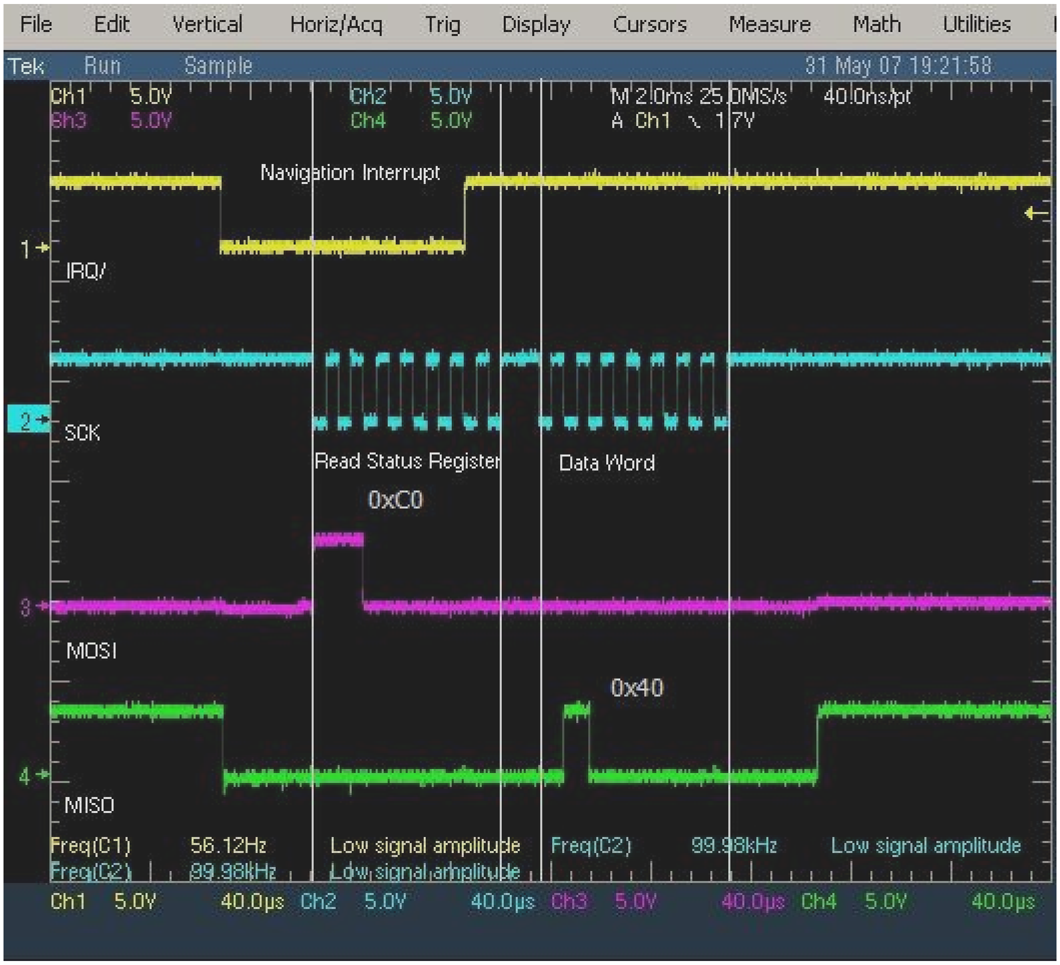

Figure 17. Navigation interrupt and read status register

When the finger starts to move over the sensor, the X and Y movement values are stored into the `NAVIGATION` registers. They are initialized after each reading. The registers only represent actions that have occurred since the last data packet sent to the host. This value is compared to the X and Y movement limits pre-programmed in the `MOVCTRL` register. When it equals this value, the sensor generates a Navigation interrupt. The Blackfin processor performs a read of the `STATUS` register to detect the sensor event. The sensor de-asserts the `/IRQ` signal upon the read of a `STATUS` register. It drives the value of `STATUS` register on `MISO` – the value indicating a navigation interrupt.

After the navigation interrupt is detected by the Blackfin processor, it switches the sensor to acquisition mode, disables the SPI, and de-asserts `/SSS`. The Blackfin processor then enables fast SPI mode by asserting `/FSS` logic low, enables SPI with faster baud rate, initiates the DMA descriptors, and configures for SPI DMA. The data acquisition phase is shown in Figure 18.

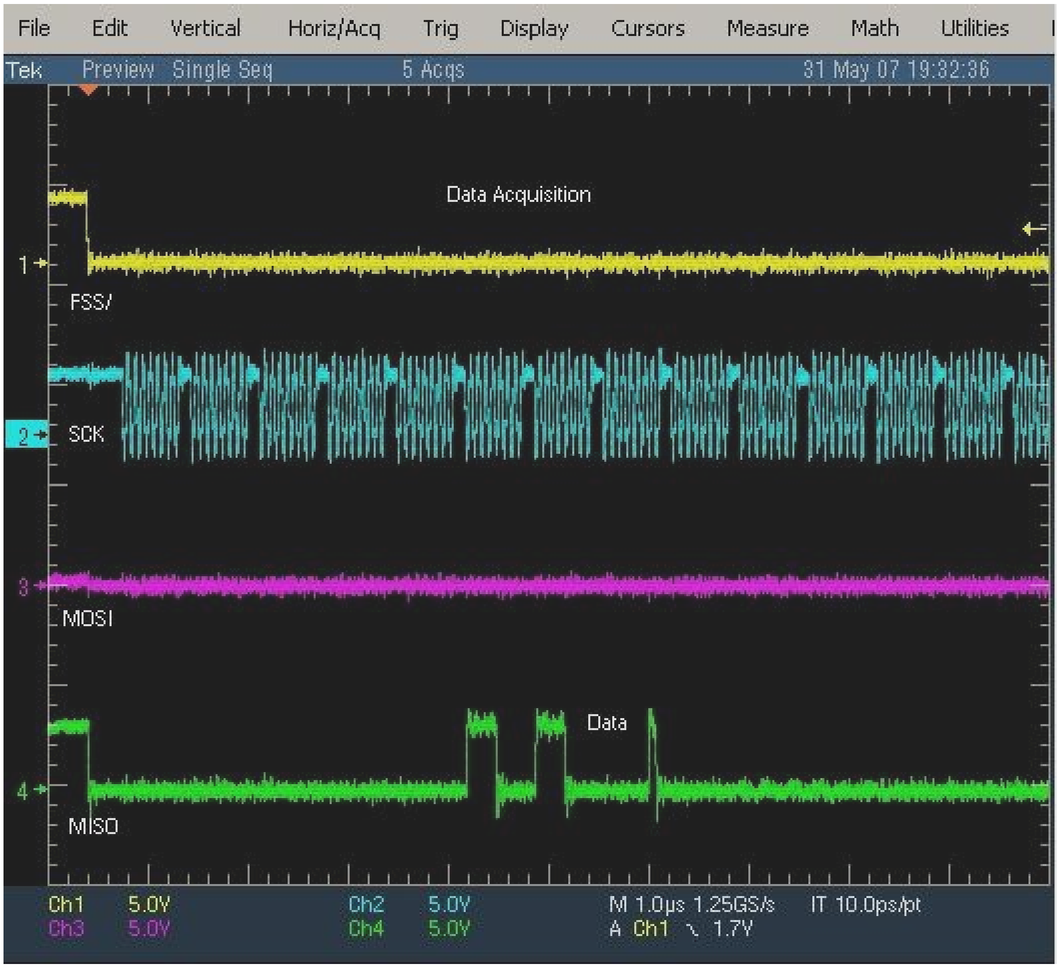

Figure 18. Data acquisition

Data Acquisition and Re-Arrangement

Acquisition mode is enabled by writing to the `MODECTRL` register. The `HEATCTRL` register, which can only be written in acquisition mode, is configured to enable heating. The watchdog timer is also enabled for a controlled heating. When heating of the sensor is requested ('1' is written in bit 6 of the `HEATCTRL` register) and the watchdog is enabled ('1' is written in bit 5 of the `HEATCTRL` register), the sensor is heated during ‘n’ seconds.

The DMA parameters are then set up for data acquisition. A large descriptor list model is used to configure the values into DMA registers. DMA flex descriptors are variable-sized data structures whose contents are loaded into DMA parameter registers. The sequence of registers in the descriptor is essentially fixed, but the length of the descriptor is completely programmable. A 2-D array, `DmaDescriptor[FN][8]` is used to configure the DMA parameters. Here, each of the 1-D arrays (`DmaDescriptor[0]`, `DmaDescriptor[1]`, `DmaDescriptor[2]`, and so on) are the individual descriptors. The first descriptor, `DmaDescriptor[0]`, is a dummy descriptor that is used to receive the first five bytes. This is needed because, for the first array or frame reading, 40 dummy clock cycles must be sent by the sensor before the first data arrives. This is necessary for the initialization of the chip pipeline.

Consequently, the first synchronization sequences appear after 40 clock cycles. For the following array readings, data arrives at every clock cycle.

The sensor sends data in the form of frames, where each frame consists of (232 + 1 dummy) x 8 pixels, each pixel being 4 bits wide. The start of each frame is marked by the dummy column, which is expected to contain a synchronization word, 0x0F0F2000. The pixel array is always read in the following order: the first byte, following the 4 bytes of the dummy column, which contains the value of the pixels physically located on the upper left corner of the array, when looking at the die with bond pads to the right. Then another 4 bytes are read that contain the value of the pixels located in the same column from top to bottom. The next column on the right is output, and so on, until the last line on the right, close to the bond pads, is output.

The data processing portion of the application performs the following functions in order to display the acquired fingerprint image.

- **Nibble-swapping:** The sensor sends data in a nibble-swapped format. For example, the synchronization word is sent as 0xF0F00200. This swapping is shown in Figure 19. The application contains a routine to swap the odd-even pixels for the entire frame.

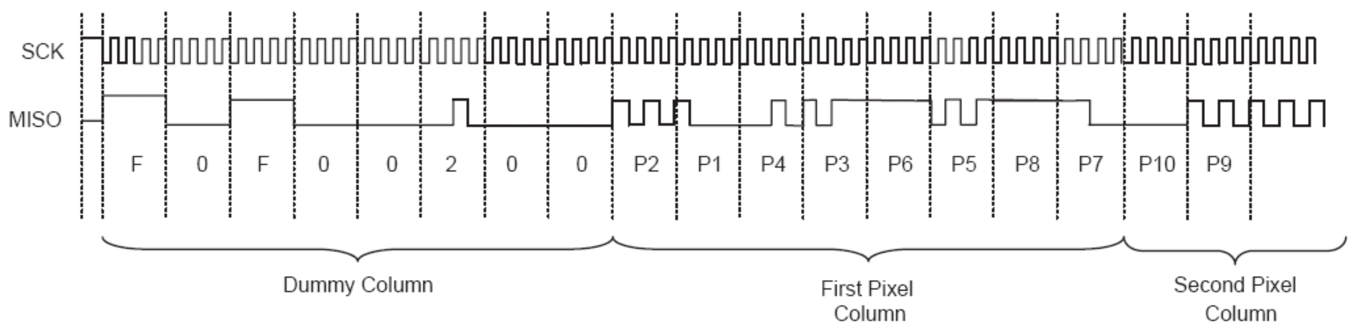

Figure 19. Reading of a frame

- **4-bit to 8-bit conversion:** Each pixel sent by the sensor is 4 bits wide, whereas the Image Viewer, or most other display applications, display images with a minimum pixel width of 8. For this reason, a zero-padding of 4 bits is done to each of the pixels so as to convert them to 8 bits wide.
- **Level adjustment:** Each pixel in the received data has an intensity in the range of 0 to 15, whereas the display range is 0 to 255. A level translation of each pixel is therefore performed to obtain a good display.
- **Array transpose:** The data is sent column-wise by the sensor. Since the 2-dimensional DMA would receive data row-wise, the data array in the Blackfin processor would be of the form Data[FN][233][4]. A transpose of this array must be taken in order to display the frames continuously. The 3-dimensional array is used here to get a continuous display of frames.

A code snippet that shows the implementation of the data-rearrangement steps described above is given in Listing 1.

```
for(k=0; k<FN; k++)
    {
        for(i=1; i<233; i++)
        {
            for(j=0; j<4; j++)
            {
                temp1 = Data[k][i][j] & 0xF;
                temp2 = Data[k][i][j] & 0xF0;
                temp2 = temp2 >> 4;
                DataProcessed[k][(j*2)][i-1] = (temp1*0xF);
                DataProcessed[k][((j*2)+1)][i-1] = (temp2*0xF);
            }
        }
    }
```

Listing 1. Data re-arrangement

An example frame is shown in Figure 20 below.

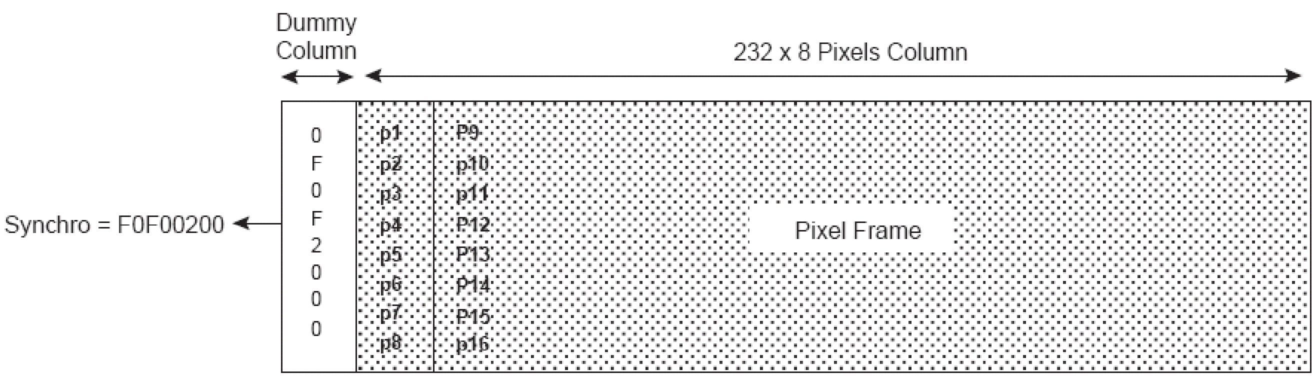

Figure 20. Example of a frame

The re-arranged data is stored into the array `DataProcessed[FN][8][233]`. This data can be viewed using the Image Viewer utility in the VisualDSP++ tools. Figure 21 shows a few sample fingerprint images acquired using the described process. The image acquired in VisualDSP++ tools, along with the settings is shown in Figure 22.

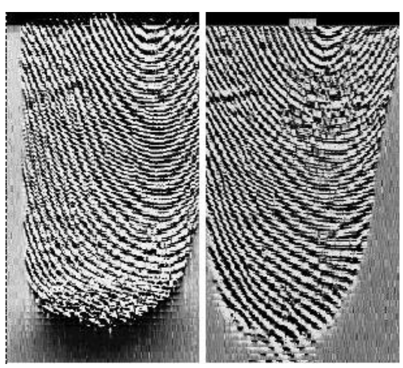

Figure 21. Examples of captured fingerprint images

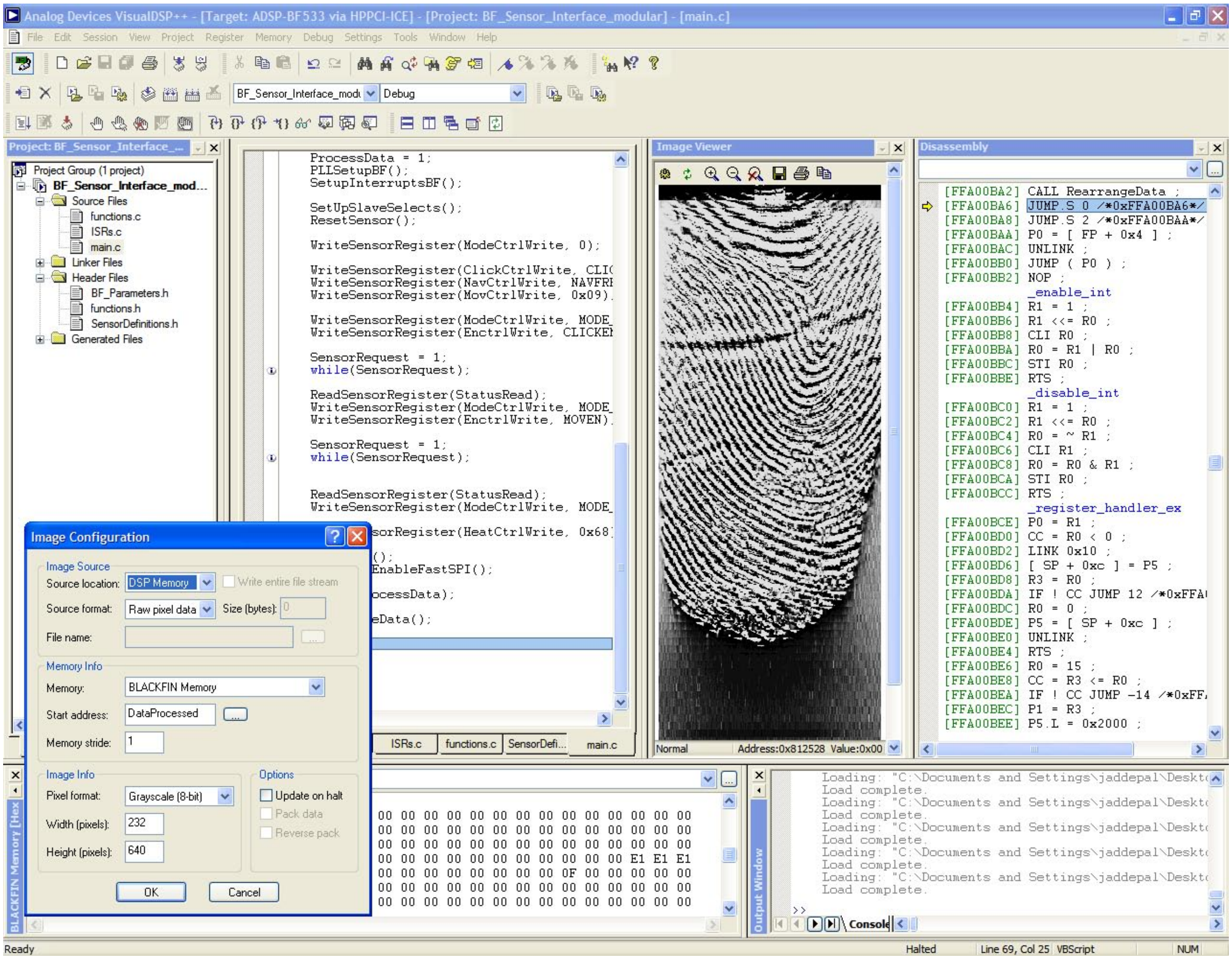

Figure 22. VisualDSP++ screen shot for image capture

Fingerprint Reconstruction and Recognition

Assuming that the fingertip is swept across the sensor window at a reasonable rate, the overlap between successive frames enables an image of the entire fingerprint to be reconstructed. This can be done using software supplied by Atmel. The reconstructed image is at 8-bit resolution due to resolution enhancement during frame reconstruction. The reconstructed image is typically 25 mm x 14 mm, which is equivalent to 500 x 280 pixels. At an 8-bit resolution per pixel, this requires 140 Kbytes of storage per image. Larger or smaller images can be derived from this using standard image processing techniques, depending on the requirements of the application.

Once the frames have been joined to obtain a complete fingerprint image, recognition algorithms are applied in order to match the sample with a template. Contact Atmel support at support.fingerchip@atmel.com for details on reconstruction and recognition software.

References

[1] *ADSP-BF533 Blackfin Processor Hardware Reference.* Rev 3.1, May 2005. Analog Devices, Inc.

[2] *Atmel FingerChip Thermal Fingerprint Sweep Sensor, AT77C104B Datasheet*, 5347D-BIOM-1/07. Atmel Corporation.

[3] *Zebra (Elastomer) Connector Types (http://www.orientdisplay.com/zebra-connector.html).* Orient Display Limited.

[4] *Atmel's FingerChip Technology for Biometric Security* - By Peter Bishop, Communications Manager. Atmel Corporation.

[5] *Atmel Applications Journal - FingerChip Technology for Biometric Security.* Atmel Corporation.

[6] *Handbook of Fingerprint Recognition*, - David Maltoni, Dario Maio, Anil K. Jain, Salil Prabhakar

Document History

Revision	**Description**
Rev 1 – August 9, 2007 *by Jayanti Addepalli and Aseem Vasudev*	Initial release