Section 6

Communication Using Serial Interfaces: UART and SPI

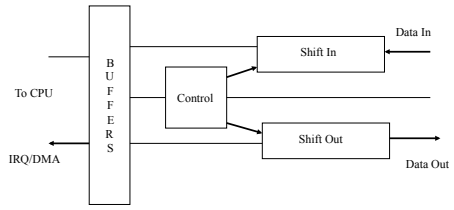
Concepts of Serial Communication

- Limitations of Parallel Bus
  - Clock skew becomes a serious issue for high speed and long distance
  - Cost of wire, fewer wires cost less and occupies less space
  - Cross talk between multiple conductors, affecting signal quality
- Serial Communication
  - Sending data a bit at a time, sequentially, over a communication channel
  - Un-clocked, no clock skew problem
  - Fewer wires → low cost, more space for better isolation from surroundings
- Simplex, half-duplex, and full-duplex
  - Simplex: data can be transferred in only one direction
  - Half-duplex: data can be transferred in two directions, but one at a time
  - Full-duplex: data can be transferred in two directions concurrently

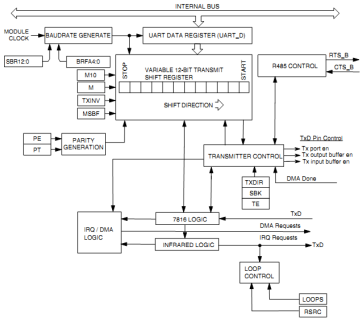
General Procedure and Logic of Serial Communication

- Special signal marks
  - Start bit and stop bit that mark the begin and end of one comm.
- Parity bit
  - Error checking and correction
    - Even parity, odd parity, CRC, Hamming code, etc.
- Baud rate
  - Specifies how fast bit sequence is transmitted
- Buffering for transmitter and receiver

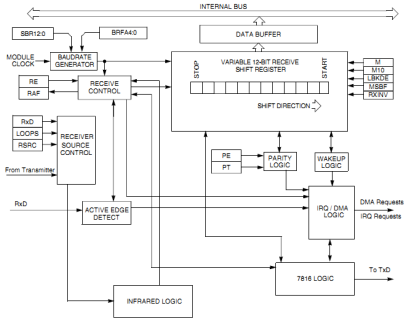
Logic Diagram of Most Serial Interfaces



UART Transmitter Block Diagram



UART Receiver Block Diagram



UART Instantiations on Kinetis

UART Instance	ISO-7816 Supported?	FIFOs	Module Clock	Maximum Baud Rate
UART0	Yes	8 entry TxFIFO 8 entry RxFIFO	Core Clock (Max freq = 100 MHz)	6.25 Mbits/sec
UART1	No	8 entry TxFIFO 8 entry RxFIFO	Core Clock (Max freq = 100 MHz)	6.25 Mbits/sec
UART2-UART5	No	No FIFOs (double buffered operation)	Peripheral Clock (Max freq = 50 MHz)	3.13 Mbits/sec

UNIVERSITY of Rhode Island

UART Detailed Signal Desciptions

Signal	I/O	Description
CTS	I	Clear to send. Indicates whether the UART can start transmitting data when flow control is enabled.
	State meaning	Asserted—Data transmission can start. Negated—Data transmission cannot start.
	Timing	Assertion—When transmitting device's RTS asserts. Negation—When transmitting device's RTS deasserts.
RTS	O	Request to send. When driven by the receiver, indicates whether the UART is ready to receive data. When driven by the transmitter, can enable an external transceiver during transmission.
	State Meaning	Asserted—When driven by the receiver, ready to receive data. When driven by the transmitter, enable the external transmitter. Negated—When driven by the receiver, not ready to receive data. When driven by the transmitter, disable the external transmitter.
	Timing	Assertion—Can occur at any time; can assert asynchronously to the other input signals. Negation—Can occur at any time; can deassert asynchronously to the other input signals.
RXD	I	Receive data. Serial data input to receiver.
	State meaning	Whether RXD is interpreted as a 1 or 0 depends on the bit encoding method along with other configuration settings.
	Timing	Sampled at a frequency determined by the module clock divided by the baud rate.
TXD	O	Transmit data. Serial data output from transmitter.
	State meaning	Whether TXD is interpreted as a 1 or 0 depends on the bit encoding method along with other configuration settings.
	Timing	Driven at the beginning or within a bit time according to the bit encoding method along with other configuration settings. Otherwise, transmissions are independent of reception timing.
Collision	I	Collision Detect. Indicates if a collision is detected during Data Transmission.
	State Meaning	Asserted—Indicates a collision detection. UARTxCPW determines the length of this pulse for valid collision detection. Negated—No collision detected.
	Timing	Asserts asynchronously to other input signals.

UNIVERSITY of Rhode Island

UART Modes of Operation

- **UART module supports three main operating modes:**
  - **UART mode**
    - **Supported on all 6 UARTs**
  - **IrDA mode**
    - **Support on all 6 UARTs**
  - **ISO-7816 mode**
    - **Only supported on UART0**

UNIVERSITY of Rhode Island

UART Mode Features

- Full-duplex serial communication
- Standard mark/space non-return-to-zero (NRZ) format
- 13-bit baud rate selection with /32 fractional divide, based on module clock frequency
- Programmable data formats
  - 8- or 9-bit data formats including 9-bit with parity
  - Programmable transmitter output and receiver input polarity
  - Ability to select MSB or LSB to be first bit on wire
- 8 entry Rx and Tx FIFOs available on UART0 and UART1
- Hardware flow control support for request to send (RTS) and clear to send (CTS) signals
- Address match feature in receiver to reduce address mark wakeup ISR overhead
- Hardware parity generation and checking
- Interfaced to the on-chip DMA
  - DMA Rx and Tx requests from the UART can be used to move data without processor intervention

UNIVERSITY of Rhode Island

IrDA Mode Features

- Selectable IrDA 1.4 return-to-zero-inverted (RZI) format
- Programmable narrow pulse transmission and detection
- Support IrDA data rates between 2.4 kbits/s and 115.2 kbits/s
- Several options for changing RX and TX sources for UART0 and UART1 are controlled by the SIM:
  - UARTn_RX input from the pin or from CMP0 or CMP1
  - UARTn_TX output directly from UART or modulated with FTM outputs
  - These options can be used in any UART mode, but are most useful for IrDA applications

►NOTE: Although all UARTs include IrDA features only UART0 and UART1 have the alternate options for the TX and RX sources controlled by the SIM

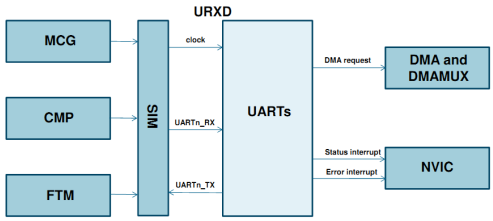
UNIVERSITY of Rhode Island

ISO-7816 Mode Features

- Support for ISO-7816 protocol for interfacing with SIM and smartcards
  - Support of T=0 and T=1 protocols
  - Automatic retransmission of NACK'd packets with programmable retry threshold
  - Support for 11 and 12 ETU transfers
  - Detection of initial packet and automated transfer parameter programming
- Interrupt-driven operation with seven ISO-7816 specific interrupts
  - Wait Time Violated
  - Character Wait Time Violated
  - Block Wait Time Violated
  - Initial Character Detected
  - Transmit Error Threshold Exceeded
  - Receive Error Threshold Exceeded
  - Guard Time Violated

UNIVERSITY of Rhode Island

UART interconnect diagram



SIM Configuration for the UART

- SIM controls clock gating for the UARTs (and other modules)
  - SIM must be initialized to enable the clock for any UART before it can be initialized (module registers are not accessible until the clock is enabled).
- SIM control TX and RX sources for UART0 and UART1
  - Default option is a direct connection between the UART and the RX and TX pins (no configuration step is needed to use the default)
  - Alternate options for RX:
    - CMP0
    - CMP1
  - Alternate options for TX:
    - UART TX output modulated with FTM1 channel 0
    - UART TX output modulated with FTM2 channel 0
- If using an alternate source for RX and/or TX, the SIM should be configured to select the source before enabling the receiver and/or transmitter in the UART

UART Operation in Low Power Modes

Low Power Mode	UART Operation	Comments
STOP	Static, wakeup on edge	UART can be a wakeup source
VLPR	Max 125kbps	Frequency is limited in VLPR mode
VLPW	Max 125kbps	Frequency is limited in VLPW mode
VLPS	Static, wakeup on edge	UART can be a wakeup source
LLS	static	UART retains state, but is not active so it cannot be used for wakeup. Several UART signals are muxed with LLWU wakeup sources though, so UART pins could potentially be used for wakeup, just not coming from the UART itself.
VLLSx	OFF	UART does not retain state. Several UART signals are muxed with LLWU wakeup sources though, so UART pins could potentially be used for wakeup, just not coming from the UART itself.

Example UART Init

1. Enable the clock for the port associated with the UART pins you want to use (SIM_MCGC5)
2. Enable UART pins (PORTx_PCRn)
3. Enable the UART module clock (SIM_MCGCn)
4. Configure the UART control registers for the desired data format
   - Number of data bits (UARTn_C1[M] and UARTn_C4[M10])
   - Parity and parity type (UARTn_C1[PE,PT])
   - MSB or LSB first (UARTn_S2[MSBF])
   - Data polarity (UARTn_S2[RXINV] and UARTn_C3[TXINV])
5. Configure the baud rate (UARTn_BDH and UARTn_BDL)
6. Enable the receiver and/or transmitter (UARTn_C2[RE, TE])

Baud Rate Calculation

- The UART has a 13-bit integer divider and a 5-bit fractional fine adjust counter that are used to generate the UART baud rate.

UART baud rate = UART module clock/ (16 * (SBR[12:0] + BRFD))

- Where BRFD = the BRFA[4:0] field divided by 32

Baud Rate Calculation Examples

The table below gives some baud rate calculation examples using a 50MHz module clock.

SBR	BRFA	BRFD	Rx Clock	Tx Clock	Target Baud Rate	Error (%)
1	0	0	50 MHz	3.125MHz	3125000	0
27	0	0	1851.85 kHz	115740.7 kHz	115200	0.47%
27	4	0.125	1843.32 kHz	115207.4 kHz	115200	0.006%
162	0	0	308.64 kHz	19290.1 kHz	19200	0.47%
162	24	0.75	307.219 kHz	19201.2 kHz	19200	0.006%
325	0	0	153.846 kHz	9615.38 kHz	9600	0.16%
325	17	0.53125	153.595 kHz	9599.69 kHz	9600	0.003%

Polling, Interrupt, or DMA configuration

- The UART can be configured to handle data flow by polling status flags, generating interrupts, or using the DMA.
- Polling is the most CPU intensive, but might make the most sense when handling small messages.
- The UART status interrupt can be used to decrease CPU loading. Status interrupt conditions are:
  - Transmit data empty
  - Transmit complete
  - Idle line (primarily used for multi-drop applications)
  - Receive data full
  - LIN break detect
  - RxD pin active edge (main use is as a CPU wakeup)
  - Initial character detect (used for ISO-7816 mode only)
- The DMA can be used to automatically move receive and/or transmit data to reduce CPU loading even more. DMA requests can be generated on:
  - Transmit data empty
  - Receive data full