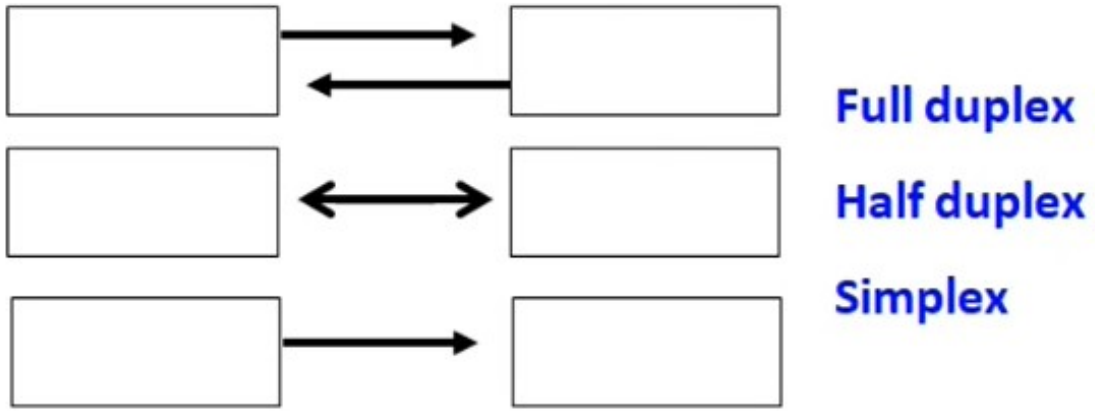
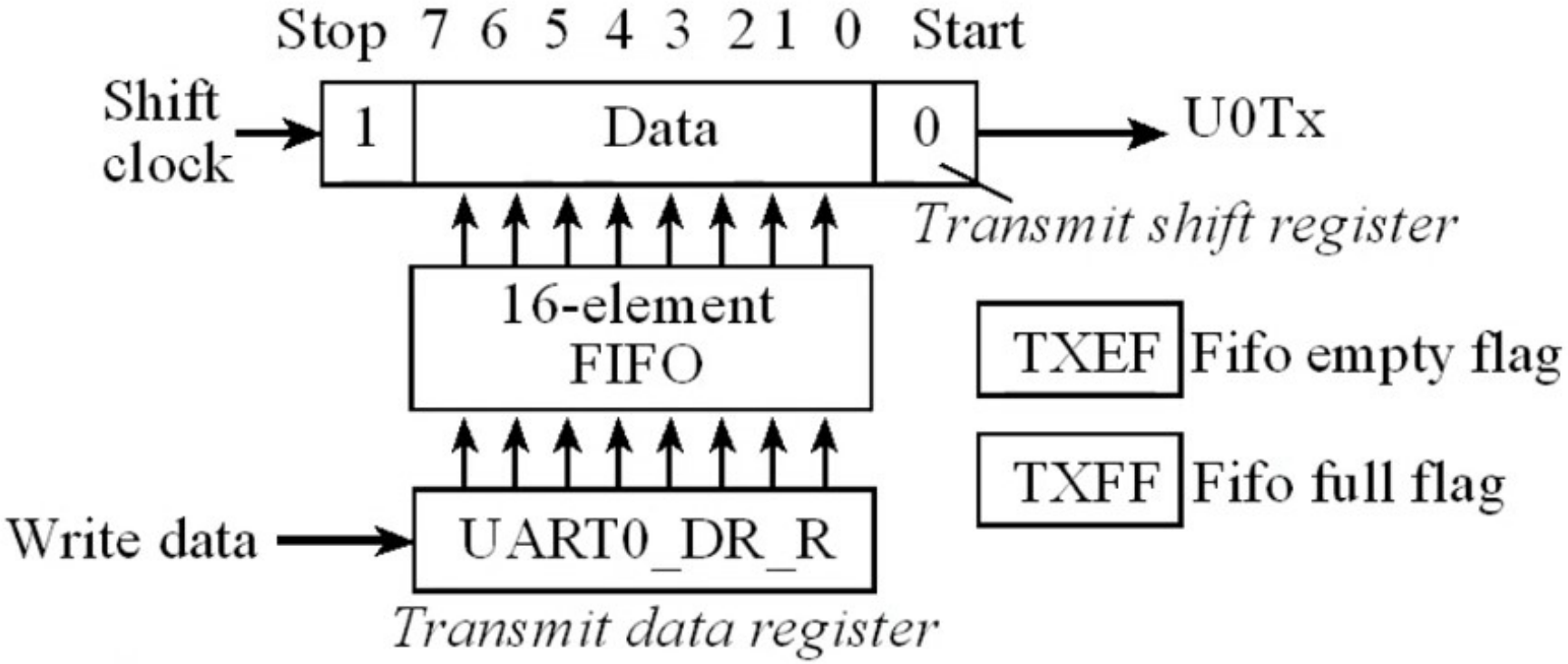
Serial I/O

- Serial communication
  - Transmit Data (TxD), Receive Data (RxD), and Signal Ground (SG) implement *duplex* communication link
  - Both communicating devices must operate at the same bit rate

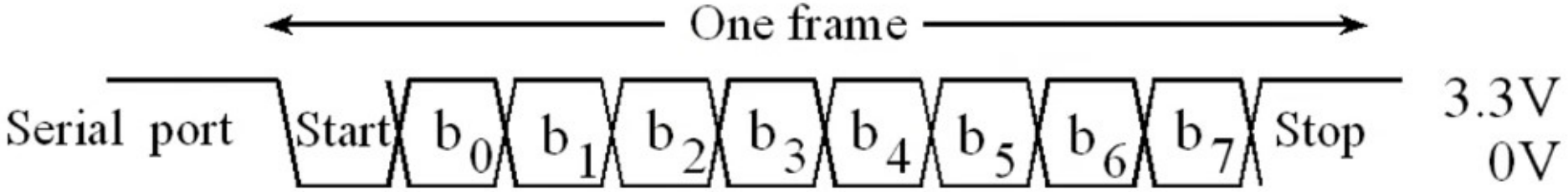


UART - Transmitter

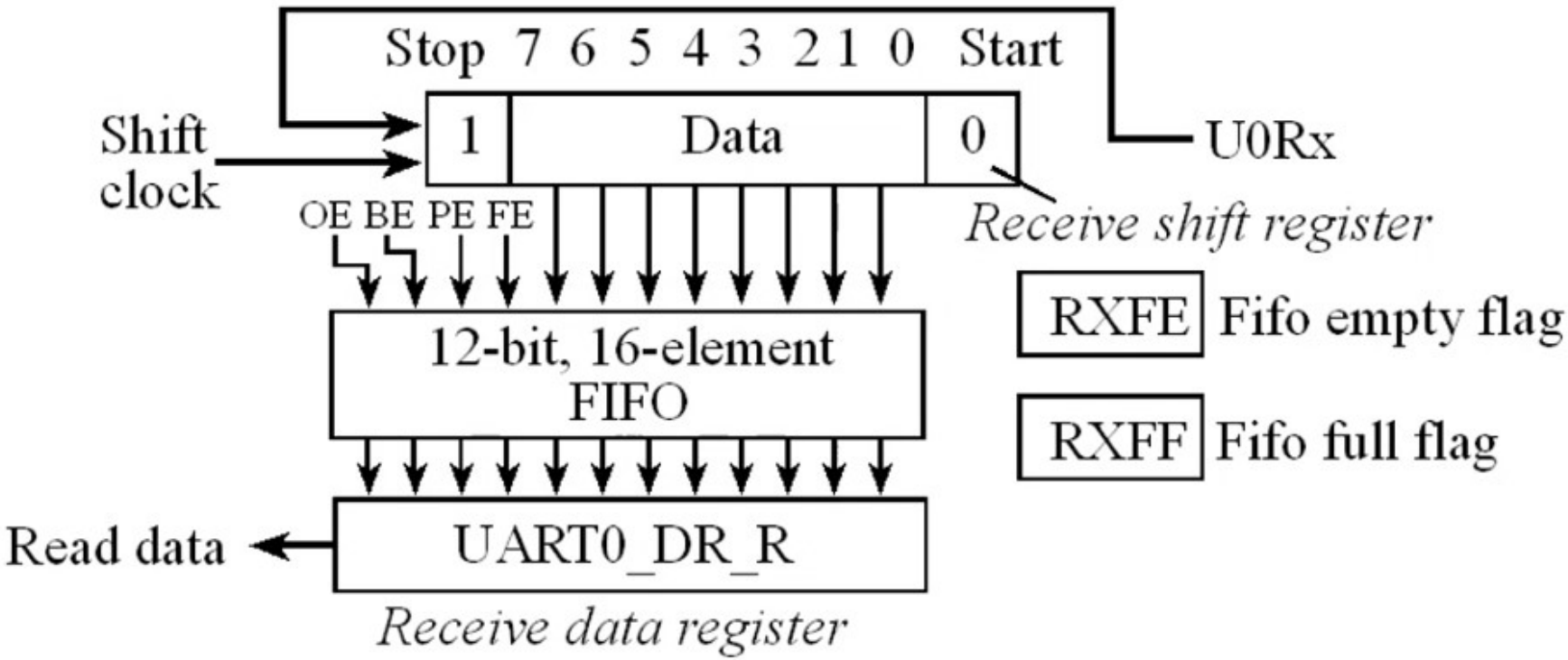


UART - Transmitter

- Tx Operation
  - Data written to UART0_DR_R
    - passes through 16-element FIFO
    - permits small amount of data rate matching between processor and UART
  - Shift clock is generated from 16x clock

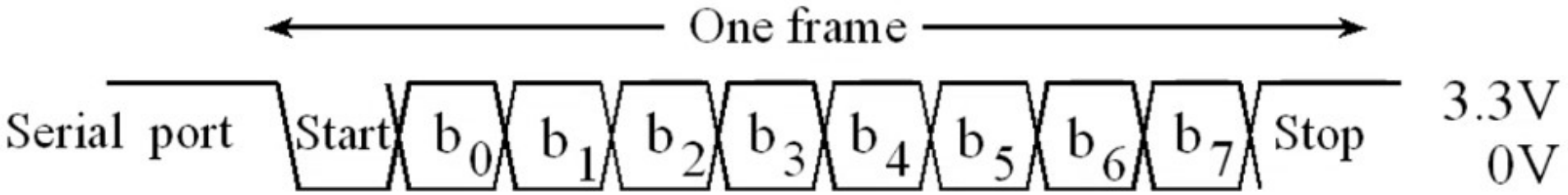


UART - Receiver

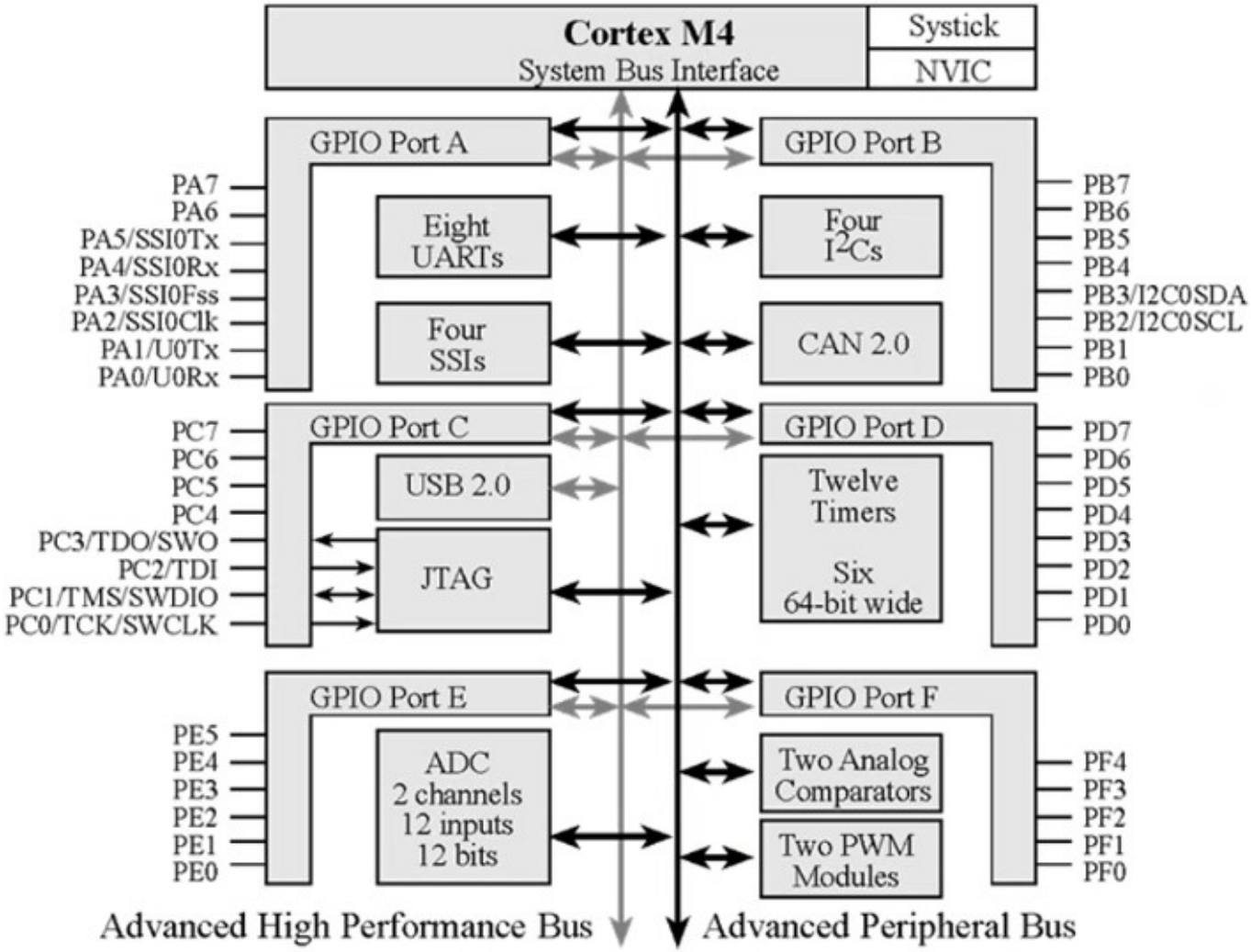


UART - Receiver

- Rx Operation
  - **RXFE** is 0 when data are available
  - **RXFF** is 1 when FIFO is full
  - FIFO entries have four control bits
    - **BE** set when Tx signal held low for more than one frame (break)
    - **OE** set when FIFO is full and new frame has arrived
    - **PE** set if frame parity error
    - **FE** set if stop bit timing error



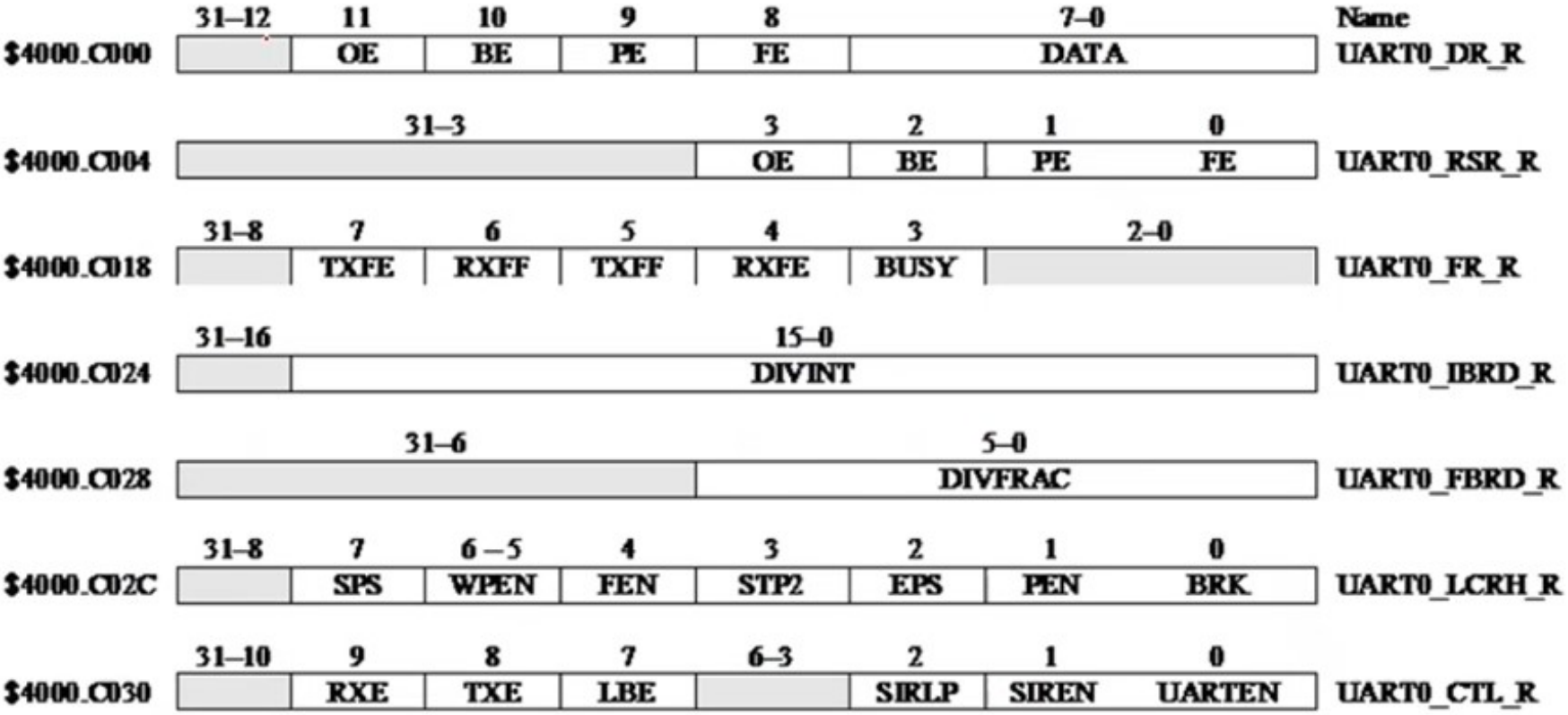
Texas Instruments TM4C123



GPIOPCTL Register

IO	Ain	0	1	2	3	4	5	6	7	8	9	14
PA2		Port		SSI0Clk								
PA3		Port		SSI0Fss								
PA4		Port		SSI0Rx								
PA5		Port		SSI0Tx								
PA6		Port			I_2C1SCL		M1PWM2					
PA7		Port			I_2C1SDA		M1PWM3					
PB0		Port	U1Rx						T2CCP0			
PB1		Port	U1Tx						T2CCP1			
PB2		Port			I_2C0SCL				T3CCP0			
PB3		Port			I_2C0SDA				T3CCP1			
PB4	Ain10	Port		SSI2Clk		M0PWM2			T1CCP0	CAN0Rx		
PB5	Ain11	Port		SSI2Fss		M0PWM3			T1CCP1	CAN0Tx		
PB6		Port		SSI2Rx		M0PWM0			T0CCP0			
PB7		Port		SSI2Tx		M0PWM1			T0CCP1			
PC4	C1-	Port	U4Rx	U1Rx		M0PWM6		IDX1	WT0CCP0	U1RTS		
PC5	C1+	Port	U4Tx	U1Tx		M0PWM7		PhA1	WT0CCP1	U1CTS		
PC6	C0+	Port	U3Rx					PhB1	WT1CCP0	USB0epen		
PC7	C0-	Port	U3Tx						WT1CCP1	USB0pflt		
PD0	Ain7	Port	SSI3Clk	SSI1Clk	I_2C3SCL	M0PWM6	M1PWM0		WT2CCP0			
PD1	Ain6	Port	SSI3Fss	SSI1Fss	I_2C3SDA	M0PWM7	M1PWM1		WT2CCP1			
PD2	Ain5	Port	SSI3Rx	SSI1Rx		M0Fault0			WT3CCP0	USB0epen		
PD3	Ain4	Port	SSI3Tx	SSI1Tx				IDX0	WT3CCP1	USB0pflt		
PD6		Port	U2Rx			M0Fault0		PhA0	WT5CCP0			
PD7		Port	U2Tx					PhB0	WT5CCP1	NMI		
PE0	Ain3	Port	U7Rx									
PE1	Ain2	Port	U7Tx									
PE2	Ain1	Port										
PE3	Ain0	Port										
PE4	Ain9	Port	U5Rx		I_2C2SCL	M0PWM4	M1PWM2			CAN0Rx		
PE5	Ain8	Port	U5Tx		I_2C2SDA	M0PWM5	M1PWM3			CAN0Tx		
PF0		Port	U1RTS	SSI1Rx	CAN0Rx		M1PWM4	PhA0	T0CCP0	NMI	C0o	
PF1		Port	U1CTS	SSI1Tx			M1PWM5	PhB0	T0CCP1		C1o	TRD1
PF2		Port		SSI1Clk		M0Fault0	M1PWM6		T1CCP0			TRD0
PF3		Port		SSI1Fss	CAN0Tx		M1PWM7		T1CCP1			TRCLK
PF4		Port					M1Fault0	IDX0	T2CCP0	USB0epen		

TM4C UART0 – Registers



TM4C UART Setup

- UART0 operation
  - UART clock started in **SYSCTL_RCGCUART_R**
  - Digital port clock started in **SYSCTL_RCGCGPIO_R**
  - **UART0_CTL_R** contains UART enable (UARTEN), Tx (TXE), and Rx enable (RXE)
    - set each to 1 to enable
    - **UART disabled during initialization**
  - **UART1_IBRD_R** and **UART1_FBRD_R** specify baud rate
    - bit rate = (bus clock frequency)/(16*divider)
    - ex: want 19.2 kb/s and bus clock is 80 MHz
    - 80 MHz/(16*19.2 k) = 26.04167 = 11010.000011_2
    - Tx and Rx clock rates must be within 5% to avoid errors
  - **GPIO_PORTC_AFSEL_R** to choose alternate function
  - **GPIO_PORTC_DEN_R** Enable digital I/O on pins 1-0
  - **GPIO_PORTC_AMSEL_R** no Analog I/O on pins 1-0
  - **write to UART0_LCRH_R to activate**

TM4C UART Programming

```
// Assumes a 80 MHz bus clock, creates 115200 baud rate
void UART_Init(void){                    // should be called only once
  SYSCTL_RCGCUART_R |= 0x00000002;  // activate UART1
  SYSCTL_RCGCGPIO_R |= 0x00000004;  // activate port C
  UART1_CTL_R &= ~0x00000001;     // disable UART
  UART1_IBRD_R = 43;      // IBRD = int(80,000,000/(16*115,200)) = int(43.40278)
  UART1_FBRD_R = 26;      // FBRD = round(0.40278 * 64) = 26
  UART1_LCRH_R = 0x00000070;  // 8 bit, no parity bits, one stop, FIFOs
  UART1_CTL_R |= 0x00000001;      // enable UART
  GPIO_PORTC_AFSEL_R |= 0x30;     // enable alt funct on PC5-4
  GPIO_PORTC_DEN_R |= 0x30;       // configure PC5-4 as UART1
  GPIO_PORTC_PCTL_R = (GPIO_PORTC_PCTL_R&0xFF00FFFF)+0x00220000;
  GPIO_PORTC_AMSEL_R &= ~0x30;    // disable analog on PC5-4
}
```