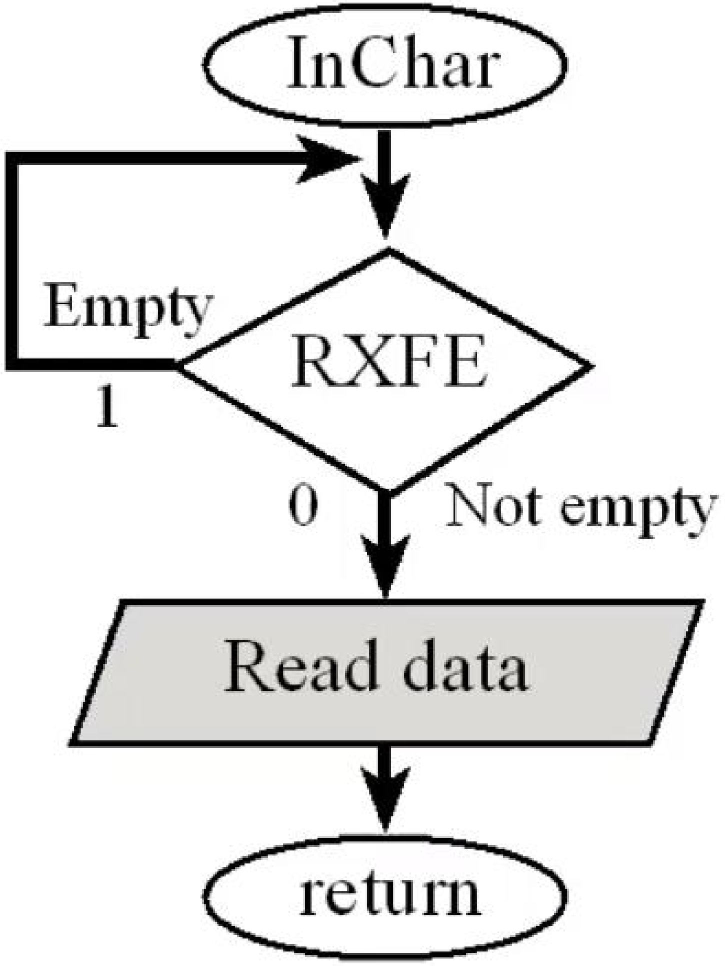
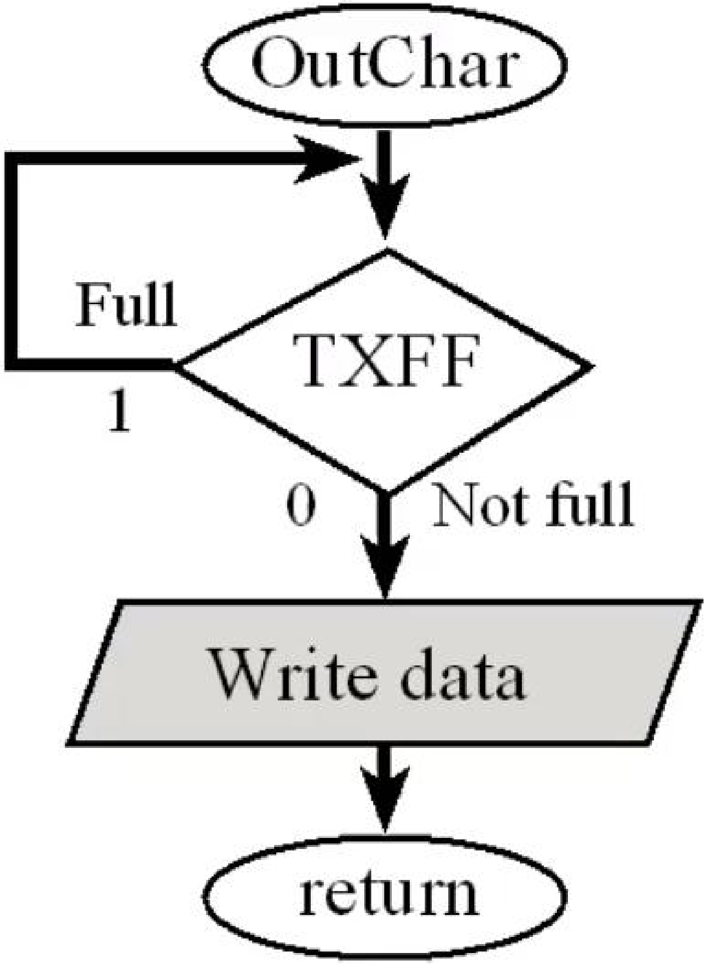
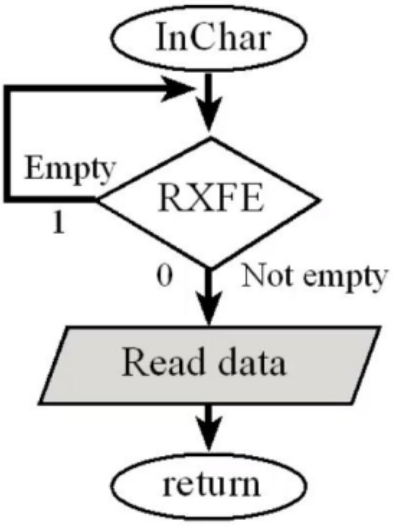
UART Synchronization

- Busy-wait operation





① Reading Data:



① Send your string as func. Parameter

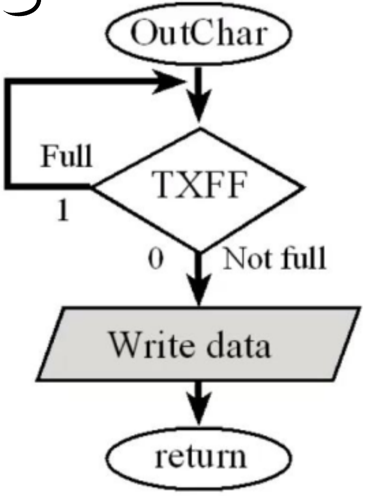
② store this charac. in UART-DR [Data]

③ before storing to Data register we need to check buffer state [FIFo] is empty ?

→ Yes: go back till be full

→ No: FIFo is full so you can pass its Data to UART Data Register and Read it

② Writting



→ like getter method in programing you need to return your character and use it in any Print func. but you first make sure about FIFO Stat is full or not

TM4C UART0 – Registers

	31–12	11	10	9	8	7–0	Name
$4000.C000		OE	BE	PE	FE	DATA	UART0_DR_R

	31–3	3	2	1	0	
$4000.C004		OE	BE	PE	FE	UART0_RSR_R

	31–8	7	6	5	4	3	2–0	
$4000.C018		TXFE	RXFF	TXFF	RXFE	BUSY		UART0_FR_R

	31–16	15–0	
$4000.C024		DIVINT	UART0_IBRD_R

	31–6	5–0	
$4000.C028		DIVFRAC	UART0_FBRD_R

	31–8	7	6 – 5	4	3	2	1	0	
$4000.C02C		SPS	WPEN	FEN	STP2	EPS	PEN	BRK	UART0_LCRH_R

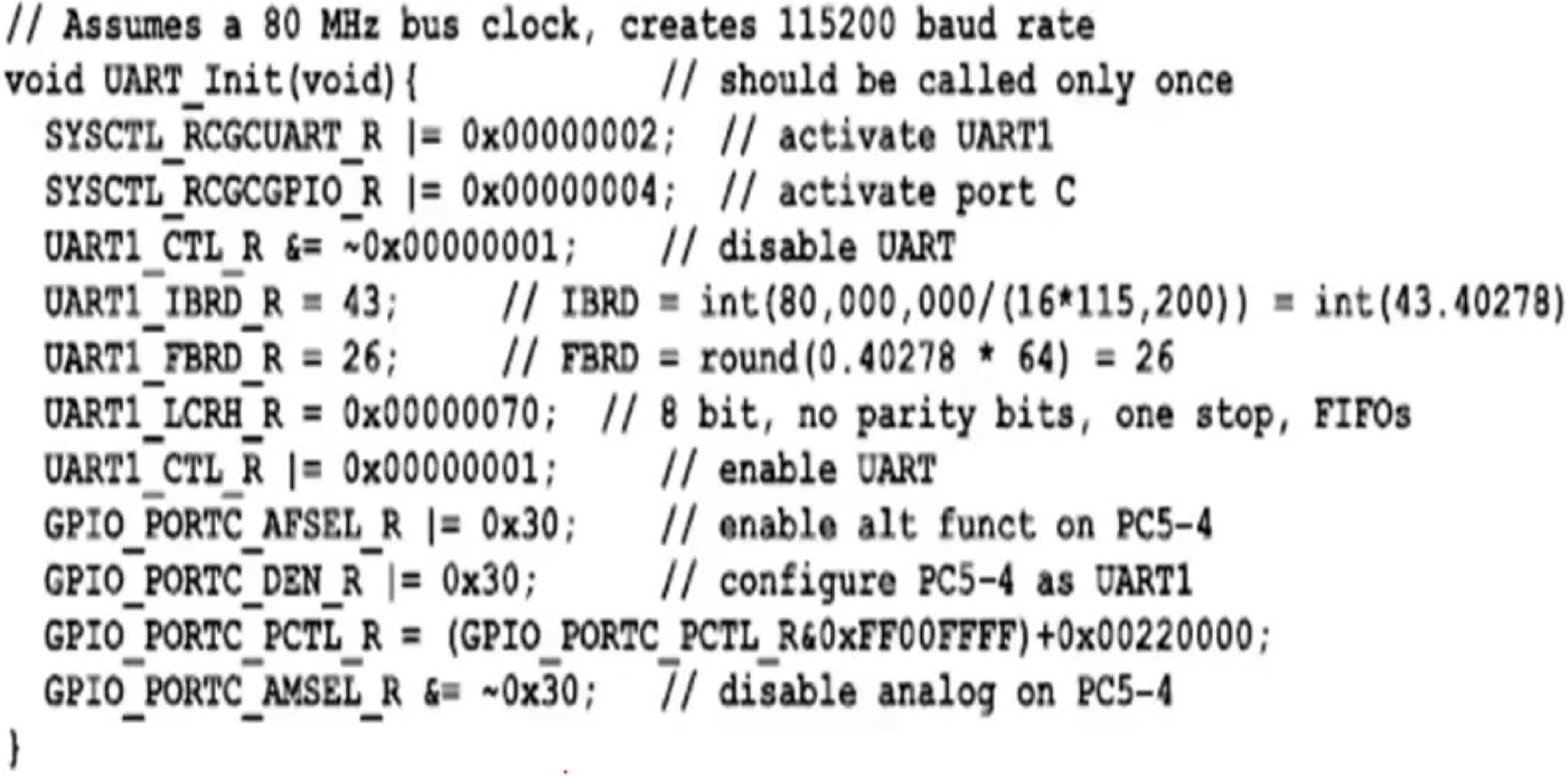
	31–10	9	8	7	6–3	2	1	0	
$4000.C030		RXE	TXE	LBE		SIRLP	SIREN	UARTEN	UART0_CTL_R

Check lect (9)

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**

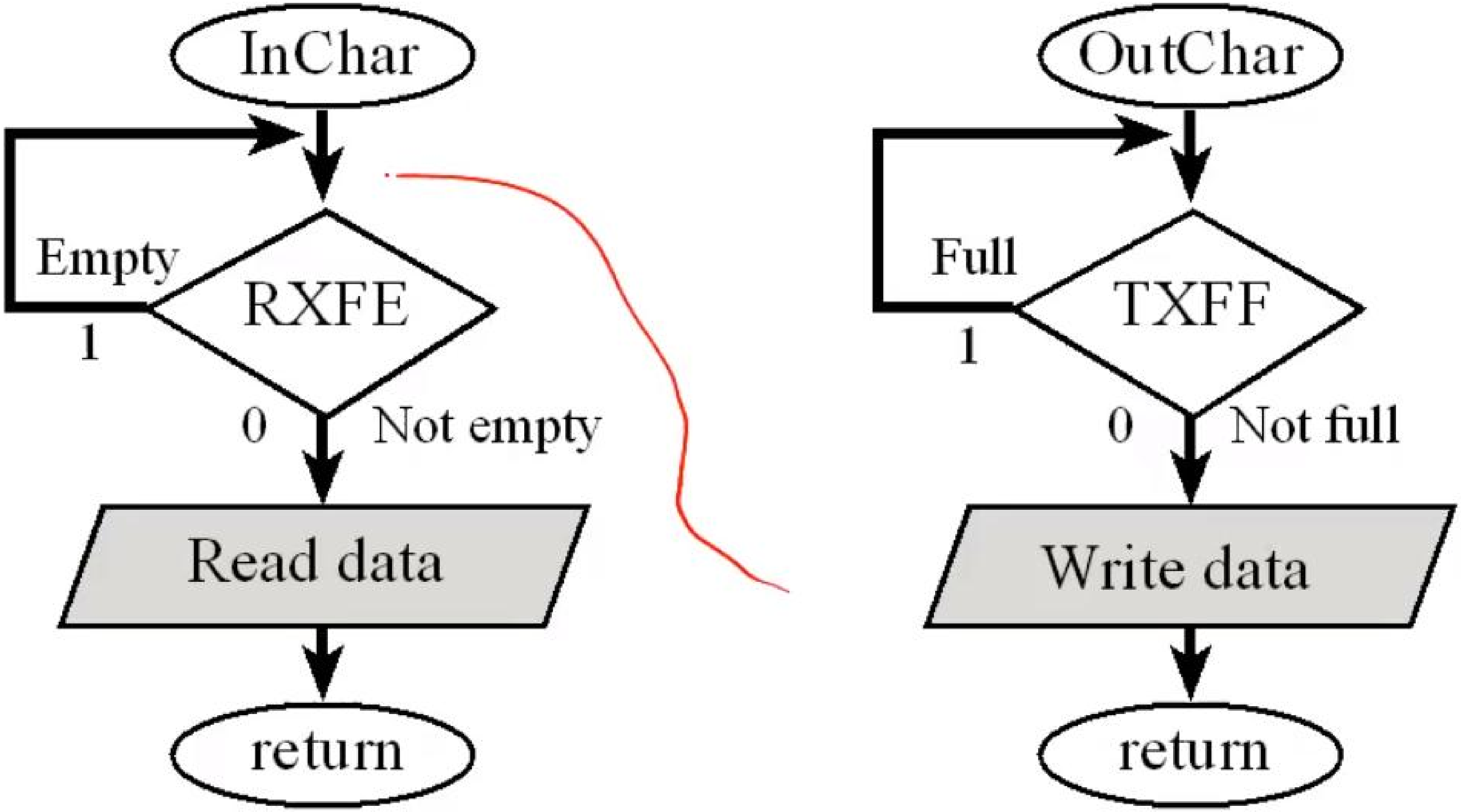
TM4C UART Programming



```
// 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
}
```

UART Synchronization

- Busy-wait operation



check 2nd slide in these slides

UART Busy-Wait Send/Recv

FR → FIFO Register

```
// Wait for new input,
// then return ASCII code
uint8_t UART_InChar(void) {
  while((UART1_FR_R&0x0010) != 0);
  // wait until RXFE is 0
  return((uint8_t)(UART1_DR_R&0xFF));
}
```

bit(4): RXFE

طالما هي مش بصفر يعني الفيفو فاضي هيستنى

إنما لو بقى مش فاضي هيعمل return

من الآخر كده لو الفيفو فاضي مفيش داتا هتتقرى

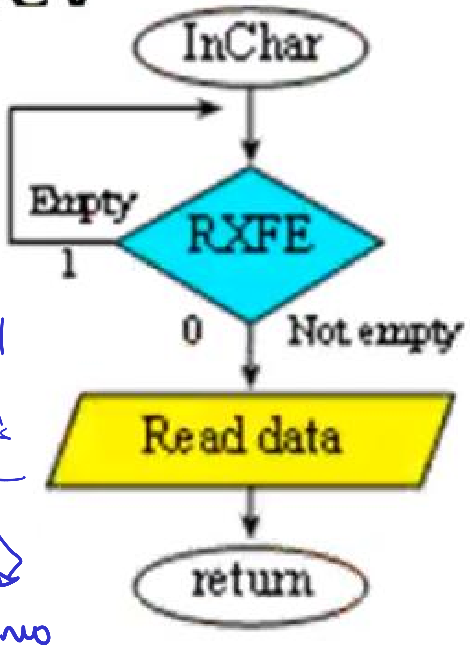
Data & FF = Data

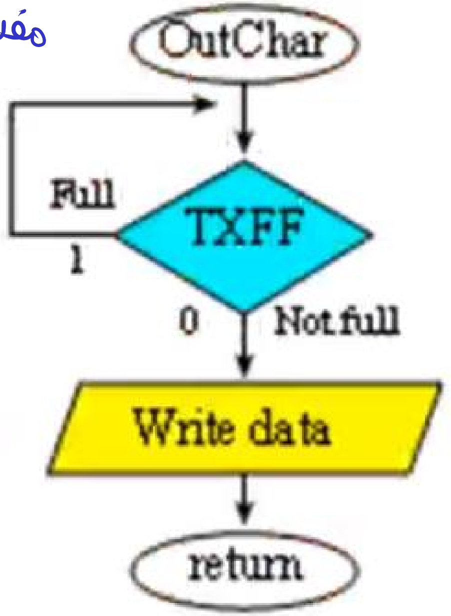
```
// Wait for buffer to be not full,
// then output
void UART_OutChar(uint8_t data) {
  while((UART1_FR_R&0x0020) != 0);
  // wait until TXFF is 0
  UART1_DR_R = data;
}
```

bit(5): TXFF:

لو الفيفو مليان مش هعرف آخد الداتا من ال UART_DR

وأبعتها على الفيفو لأن مفيش مكان


InChar
Empty
1
RXFE
0
Not empty
Read data
return


OutChar
Full
1
TXFF
0
Not full
Write data
return