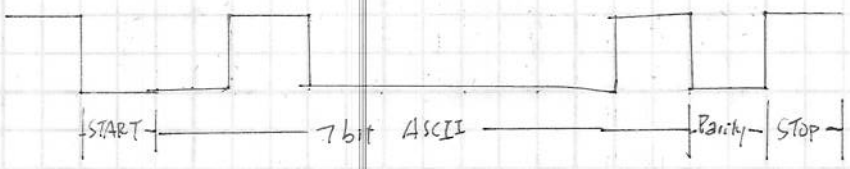
UART $42_H = 100\ 0010$

1)



2) Baud rate = 9,600 b/sec

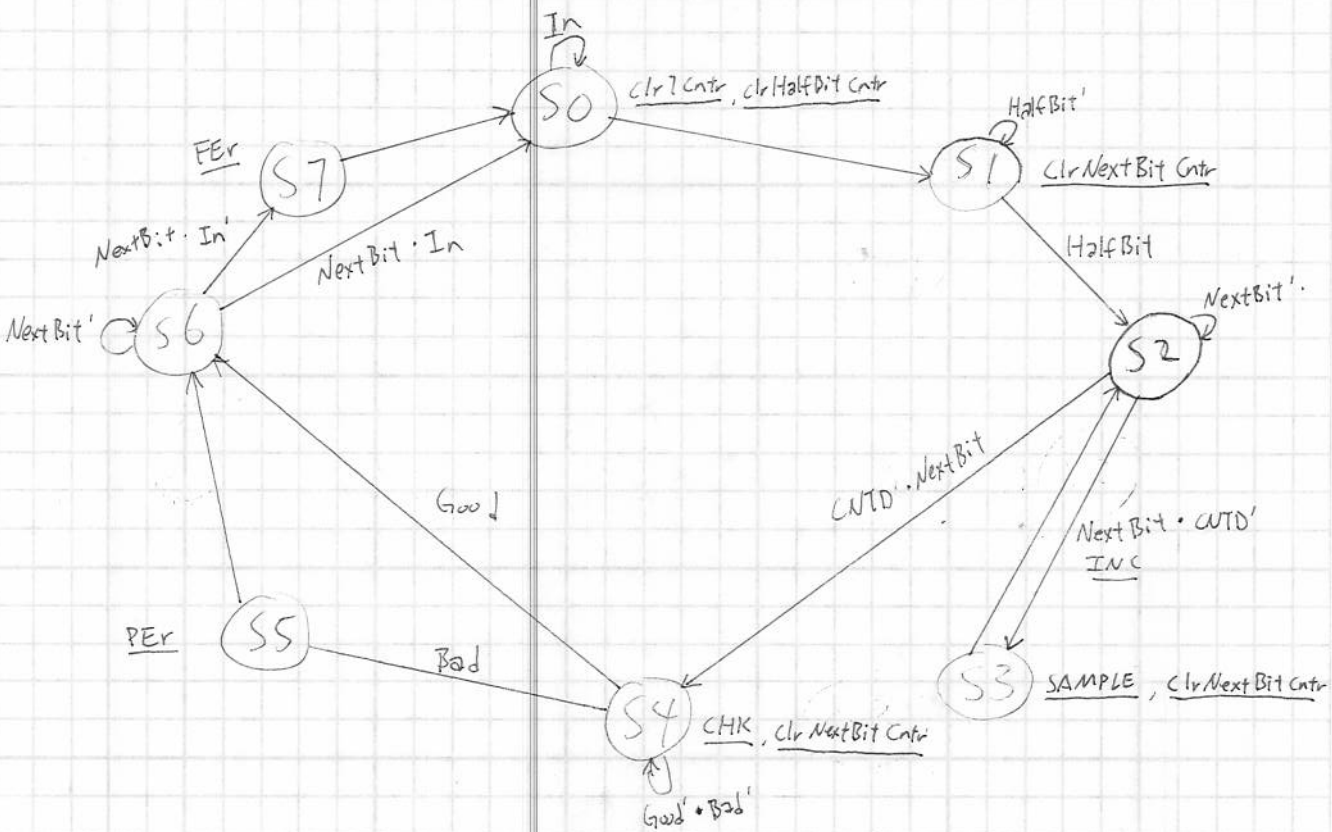
$\frac{10 \text{ bit}}{9600 \text{ b/s}} = \frac{1}{960} \text{ sec}$

$\frac{9600 \text{ b/s}}{10 \text{ bits/char}} = 960 \text{ characters/sec}$

3) Assume a counter that counts to 7 then returns CNTD

outputs: ClrHalfBit Cntr
ClrNextBit Cntr
Clr 7 Cntr
INC – increments the counter that counts to 7
CHK – tells parity checker to check
PEr – Tells host there has been a parity error
FEr – Tells host there has been a framing error
sAmple – Tells the shift registers to shift the input in.

Inputs: HalfBit
NextBit
In – the input
CNTD – counter returns this when it has reached 7
Good – asserted by parity checker, if there is no error
Bad – " " if there is an error



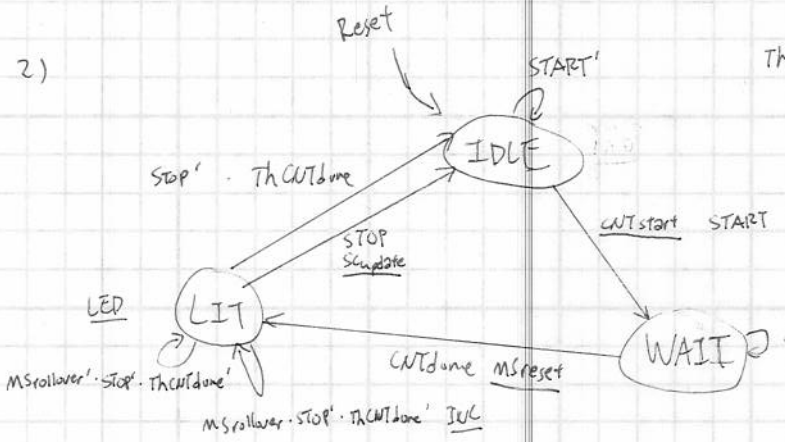
Lab 11 Prep.

1) $1\text{kHz} = \frac{1}{1000}\ \text{sec}$ cycles = 1,000,000 ns

50 MHz signal = 20 ns cycle

1000000 / 20 = 50,000 Load 49,999

2)



There's a 10,000 counter that asserts ThCNTdone when it hits 10,000. It increments when INC is asserted.

3) IDLE = 001
WAIT = 010
LIT = 100

Reset	CNT done	MS rollover	STOP	START	TH CNT done	CSTATE	NSTATE
1	–	–	–	–	–	–	IDLE
0	–	–	–	0	–	IDLE	IDLE
0	–	–	0	–	1	LIT	IDLE
0	–	–	1	–	–	LIT	IDLE
0	–	–	–	1	–	IDLE	WAIT
0	1	–	–	–	–	WAIT	LIT
0	–	–	0	–	0	LIT	LIT

N_{IDLE} = Reset + IDLE · START' + LIT (STOP + STOP · THCNTdone)

N WAIT = IDLE · START

N LIT = WAIT · CNTdone + LIT · STOP' · THCNTdone'

LED = LIT

CNT_{start} = IDLE · START

MSreset = WAIT · CNTdone

INC = LIT · MSrollover · STOP' · THCNTdone'

SCupdate = LIT · STOP