Design (VART) clk= 1MHz Bandrate 9600 clk : Single bil - Wait court Wait count 2 Input clk (freg) Baudrate Explanation: Bandrate - I bit her time unit Ibil time x ilpolk freg - wait court // Generate trigger. parameter clk-value. 100_000 parameter band - 9600 vieg bit done 20; untiger count :0; parametes idle 20; send. 1, chut 2; vug [1:0] stale : idle;

legin it (state = zidle) legin Count = 0; led. else legin if (count = zwait-count) legin lit done \(\alpha = \beta \) count \(\alpha = 0 \); end else legin lit done \(\alpha = \beta \); count \(\alpha = 0 \); end else legin it/(leachtin count \(\alpha = \count + 1 \); lit done \(\alpha = \beta \); lit done \(\alpha = \beta \); end end end there lit done acts to be as trigger. For transmission or receive: But \(\alpha \) \(\beta = \beta \); \(\alpha \) \(\beta \) \(\alpha \) \(\beta	always @(posedge clk)	
if (state = = idle) lugin Count 5); else legin if (count = = wait-count) legin lit done 2 = 11b1; count 2 = 0; else lugin wif (leacher recount +1; lottdone (= 11b0; end end end end for bransmission or recurs: Shap DO D1 D2 D3 D4 D5 D6 D3 PB stop word begin word by loc. Shap DO D1 D2 D3 D4 D5 D6 D3 PB stop word by loc. Shap Do D1 D2 D3 D4 D5 D6 D3 PB stop word by loc. Shap Do D1 D2 D3 D4 D5 D6 D3 PB stop word by loc. Shap Do D1 D2 D3 D4 D5 D6 D3 PB stop word by loc. Shap Do D1 D2 D3 D4 D5 D6 D3 PB stop word by loc. Shap Do D1 D2 D3 D4 D5 D6 D3 PB stop word by loc. Shap Do D1 D2 D3 D4 D5 D6 D3 PB stop word by loc. Shap Do D1 D2 D3 D4 D5 D6 D3 PB stop word by loc. Shap D0 D1 D2 D3 D4 D5 D6 D3 PB stop word by loc. Shap D0 D1 D2 D3 D4 D5 D6 D3 PB stop	^ //	
else legin if (court = = wait-court) legin lit done 2 = 11b1; court 2 = 0; Lend else legin vit/leselder court = court +1; lottdone 2 = 11b0; end end end for transmission or receive: Bua DO D1 D2 D3 D4 D5 D6 D2 PB Stop wards to be court = court to be court to be as trigger.	if (state = 2 idle)	If Stays at ridle until the State
else legin if (court = = wait-court) legin lit done 2 = 11b1; court 2 = 0; Lend else legin vit/leselder court = court +1; lottdone 2 = 11b0; end end end for transmission or receive: Bua DO D1 D2 D3 D4 D5 D6 D2 PB Stop wards to be court = court to be court to be as trigger.	legin	changes to check or viewire.
else legin if (court = = wait-court) legin lit done 2 = 11b1; court 2 = 0; Lend else legin vit/leselder court = court +1; lottdone 2 = 11b0; end end end for transmission or receive: Bua DO D1 D2 D3 D4 D5 D6 D2 PB Stop wards to be court = court to be court to be as trigger.		Excount Stays tube on O.
else legin if (count = = wait-count) legin wit done \(\frac{2}{2} \) 1/b ; count \(\frac{2}{2} \) 0; dend else begin with (lese to recount +1; bit done \(\frac{2}{2} \) 1/b0; end end end end end end end en	▲	
legin lit done 2 = 11b1; Count 2 = 0; Let begin let begin let begin let begin let begin let begin count = Count +1; bit done 2 = 11b0; Lit done 2 = 11b0; Lit done 2 = 11b0; Lit done alt to be as trigger. For transmission or receive: Parity lit The plays 0. Parity lit The plays 1. Read DO DI D2 D3 D4 D5 D6 D4 PB stop Stop logic 1. The plays 1. The plays 1. Read DO DI D2 D3 D4 D5 D6 D4 PB stop Stop logic 1.	A A	
lit done 12 11b1; Count 2= 0; Level by the lit transmission is success & again the count terms to be 0. Level begin Level by the increments the count variable lill the value weather waters watered the value washes wait count, where litedone 8 hays 0. Level bit done acts to be as trigger. Level by the bit done acts to be as trigger. Level by the bit of the sort washington or receive: Level by the bit of the sort washington or receive: Level by the bit of the sort washington or receive: Level by the bit of the sort washington or receive: Level by the box of the sort washington or receive: Level by the box of the sort washington or receive: Level by the box of the sort washington or receive: Level by the box of the count that the count washington washington to be a sort to be a sort washington or received. Level by the box of the count washington		Y
count 2=0; Lever begin Let a begin Let a begin Count 4= Count +1; Loit done 2= 11b0; Lit done Shays 0. Lend Lit done acts to be as brigger. For transmission or receive: Parity lit This increments the count value weathers wait count value weathers wait count, where weathers wait count, where here hit - done acts to be as brigger. Lever for transmission or receive: Lever for t	begin	After Sucurful waik count
else begin It a increments the count Variable till the value veralls waste wait count, where bit done (= 1160; end end end end there bit done acts to be as trigger. For transmission or receive: Base DO DI D2 D3 D4 D5 D6 D4 PB stop worddate 1 This increments the count variable till the value veralls till the value veralls till done acts to be as trigger. For transmission or receive: Parity bit This prophagical till the page of	bit done 22 1161;	the bit transmission is
else begin ***Xt! Cleachter* Count <= Count +1; Varnable lill the value vealus water stays one and end there lite done acts to be as trigger. For transmission or receive: Party lite	count L= 0;	turns to be o.
count (= count +1; Variable bill the value vealues wastrount, where bit done stays 0. end end end there bit done acts to be as trigger. For transmission or receive: Parity bit Bas DO DI D2 D3 D4 D5 D6 D7 PB stop wordplate The propherical state of the propherical stop by the propheric	end.	
Count $l = lount + 1$; Variable lill the value weather waite out to be value of the lit done stays 0. end end end there lit done acts to be as trigger. For transmission or receive: Parity lit There lit stop logic 1 Shap DO DI D2 D3 D4 D5 D6 D4 PB stop Shap DO DI D2 D3 D4 D5 D6 D4 PB stop Shap DO DI D2 D3 D4 D5 D6 D4 PB stop Shap DO DI D2 D3 D4 D5 D6 D4 PB stop Shap DO DI D2 D3 D4 D5 D6 D4 PB stop Shap DO DI D2 D3 D4 D5 D6 D4 PB stop	else begin	I the increme to the count
bit done (= 1160; shows to be done at to be as trigger. For transmission or receive: Baa DO DI D2 D3 D4 D5 D6 D7 PB stop Wordplate + T T T Stop logic 1	It Coulder	. Variable lill the value
end end there lit_done acts to be as trigger. For transmission or receive: Parity lit Saa DO DI D2 D3 D4 D5 D6 D7 PB stop wordplate to I I I I shop logic 1	count = count	viales wait court, where
For transmission or receive: Parity lit	Lot dime (2 100)	(/
For transmission or receive: Parity lit	eng	. Here bil - done acts to be
For transmission or receive: Parity lit		as trigger.
Sha DO DI D2 D3 D4 D5 D6 D7 PB stop Worddate + T T T Shop logic 1 Sample stop	•	
Sha DO DI	For framemission or receive:	Parity bit
La condidate The plane 1		D4 D5 D6 D7 PB STOP
The complestop	20 1 1	
T Complestop	The tolde	late of 17 17 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
Start by Incoming data sampled at Sample stop detailing transition but pulse center from logic 1 to 0		
Stort dy Incoming data sampled at but dutuling transition but pulse center from logic 1 to 0	1	1 Samplestop
from logi 1 to 0	Start by Incomin	g date sampled at but
	from logi 1 too	voit privat centur
		*

Mransmission loga. oreg [9:0] +x Data; integer dit Index = 0; vieg [9:0] shift = 0; 11 fordebug always @ Coosege clk) lugin Case (State) idle: begin tx (= 11bl; tx Data = 0; bit Index (=0; Shifttx=0; it (Start == 1161) begin Ex Datel & [1'b], txin, 1'boy; State & Send . Stap Lyth Stan State & Send; else begin State = idle ; Send: begin Exc= Ex Data [Soit Index], State Echeck; shift <= { tx Data [bit Index], Shiftex [9:1] }; end. Check: begin if (dit Index 2 2, 9) begin if (bit Bone = 1161) begin begin State L 2 send; bit Inden (= bit Index +1;

else begin 8 kete Gidle; bit Indea L20; default: State (= idle end case 14 bit done 2211b1) 9 11b1: 11b0; assign Exdone 2 (dit Index 22 9 11 Leceierer logie integer rount : 0; integer rinder = 0; parameter ridle = 0, rwait = 1, recy = 2, rcheck=3; vieg [1:0] vstate; reg [9:07 rxdata; always @ (posedge clk) legin (rstate) vidle: begin yx date (= 0; vinder (= 0; recount L20; if (8x 2 2 11b0) begin retate 12 rwait; begin rstate 12 vidle; and rwait: begin if (recount & wait count/2) begin 1/sampling redate of recount 2 recount +1; modelle of bit rstate 22 rwait;

begin recount (= 0; rstate = reu; redate (= frx, rx data [9:1]); und veey: begin it (rindex L 29) begin if (bit Done 22/161) begin rinder (= rinder +1) rstate 12 rwait; else begin rstele = ndle; r onder (=0; defaull: rstele = vidle; endcase assign rout: redata [8:1]; ordone = (rondex = 29 28 bit done = 21.61) ? 1.61:1.60); end module Ports for the module. (input clk, Soft, infect [7:0] truin, output tx, input Tx, output Tx out, output rx done, tredone);