Dataflow model verilog code for simple sop equation Y= (AB)+(co) module solequation (A, B, c, D, Y); input A, B, C, D; output Y; assign 4 = (A+B) + (C+D); endmodule Verilog cope for 4 to 1. Mux using data flow modeling module muxital (A, B, Do, D, Dz, Dz, Yz); input A, B, Do, Dr, D2, D3, b assign V = A? (B? D3: D2): (B? D, : Do); endmodule Verilog code for 3 to 8 decoder using data flow modeling module decoder (output [0:7]D, input A/B, c, endble); . drin cossign DEO] = (~ ARNBENC, & enable), .... D[1] = (NA & NB & C. f. enable), D[2] = (NA + B & N C) f enable), 0[3] = (NA + B & C & enable), D[4] = (AfNBfine fenable), in which walling D[5] = (A & NB & C) enable), D[6] = 11(A & B & NC & enable), D(+) = (A + B & confenable); end module 90 150 years Ex. (12 A Verilog vode for 2 to 1 Mux using data flow model > module sto1Mux (A, D, Do, Y); input A, D, Do; output Y; ausign y = A?(Di: Do); endmodule.

Verilog water of atom Mux using behavioural model (if-else statements) + module sto1 Mux (A, Do, D, y); input A, Do, Di; output Y; always @ (A or Do or Di) ij (A = = 1) Y = 0, 3 else Y = Do; endmodule. Using case statement: module atoMux (A, Do, D, Y); Input A, Do, Di; output Y; always @ (A or Do or Di) case (A) 0: Y = Do, 1 : Y = Di; endcase: endmodule Verilog vode for 4to1 mux using behavoral modeling. module 4tolmux (A, B, Do, D, D, D, D, Y); input A, B, Do, D, , Dz, D3: output in; where had a regular roughly an a labore stege of the least of the state of the state of the always @ (A or B por Do or D, or D2 or D3) case ((A, By) 1/ toncatenation of A and B\*/ 0: Y = Do; 1 : Y = . Dist 2: Y = D; . 3: 4= D3 9 10 1 endmodule.

Verilog vode for demux1 to 4 using behavioural model / 2 to Adecado module demuxto4 (A, D, Y); Proput [1:0] A; input D; output = [3:0] Y; req [3:0] Y; always @ (A or D) case ( { D, Ay) // concentation of A and D and DES MSB 3' b100 : Y = 4' b 0001; 3 b101 : Y = 4 b0010; 3' b110 : Y = 4'b0\$00; 3' bill : Y = 4'b1000'3 default: Y = 4'60000°, endoue endmodule Virilog code for Breatch. module sklatch (s, R, EN, 9); input S, R, EN; outfut 9; always @ (EN Ors or R) if clending = sund (MRI4 g) in x marsh a market chosed D FF positive jedge triggerd and negative. module DFFpas (D, C, 9); module DFFneg (D, C, 9); input D, C; 11-Cis clock is input D, C; outfut 9 3 de brom A po suissement outfut (95, A de 120 rieg 9; eleg of ; - 1 always @ (posedge c) always a (negedge () 9=D; endmodule endmodule

clocked DFF :module DFF (D, Ed, q); input D, EN; // corpora output 9; meg g; always @ (EN or D) 9 (EN = = 1) 9 = D; end module clocked SR FF:module SRFF (S, EN, R, g) Proput & SREN; always @ (EN or 9 or R) \$ (EN = =1) g= s | (~R + q); endmodule - Me 10/0 10001 100 Verilog wide that converts D FF to SR FF module SRFFm(S, R, c, g); input S,R,C; // cis clock output 9: 姓 章 of his of in half assigna DSR = SI (NR &g); / characteristic egn for SR FF DFFneg D1 (DSR, C, g) in Minitating neg edge trigg DIFF module Diffreq (D, 6,9); "Input Due; insternation in the forms assured out at 193 a million par some bais rieg 9; always @ (negedge () endmodule