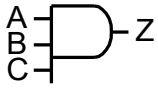
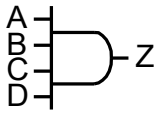
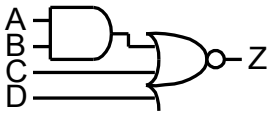
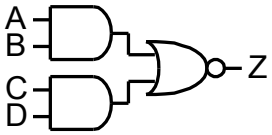
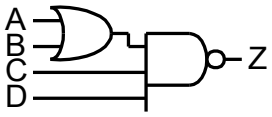
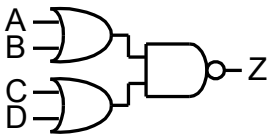
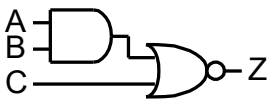
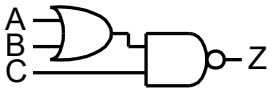
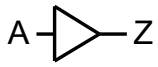
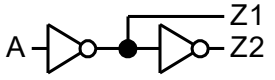
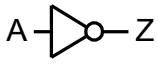
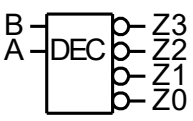
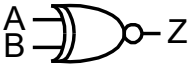
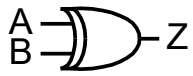
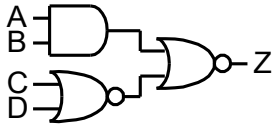
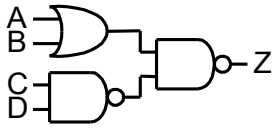
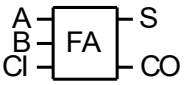
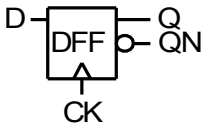
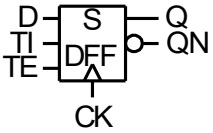
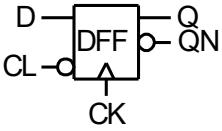
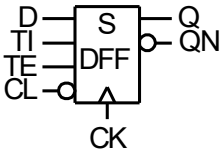
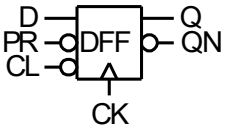
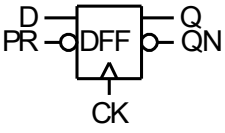
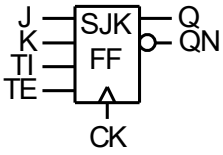
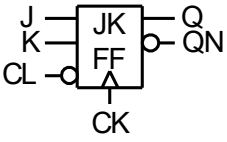
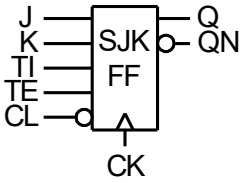
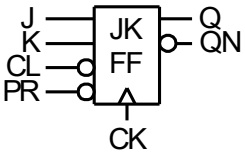
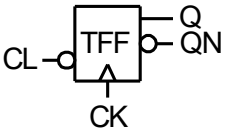
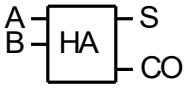
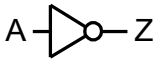
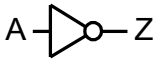
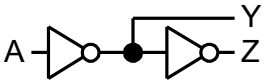
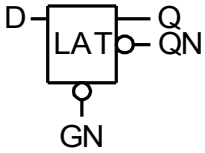
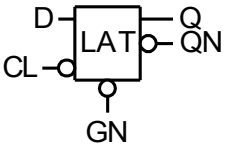
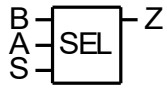
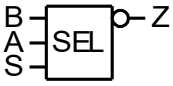
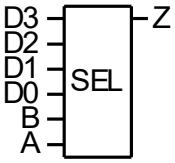
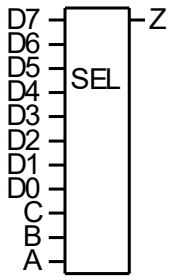
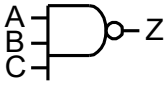
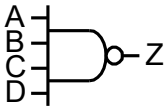
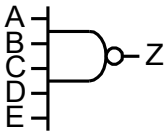
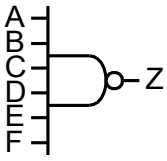
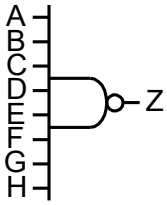
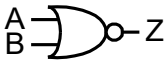
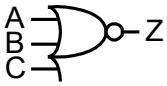
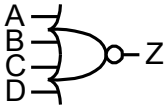
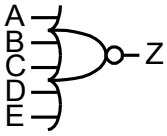
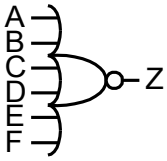
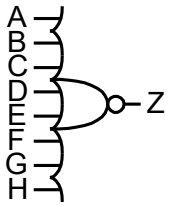
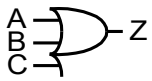
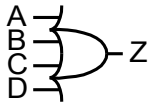
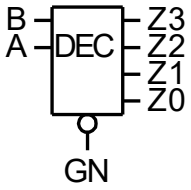
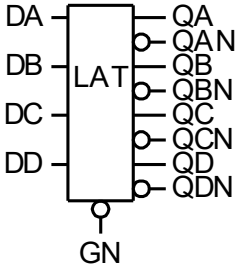
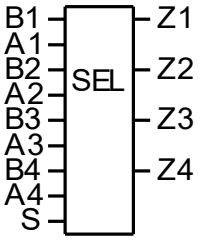
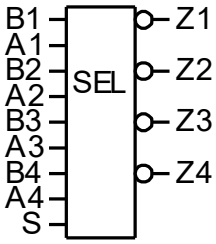
Toshiba TC110G Macrocells on 82C457

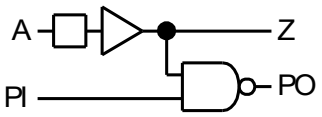
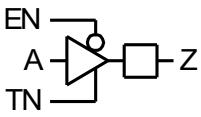
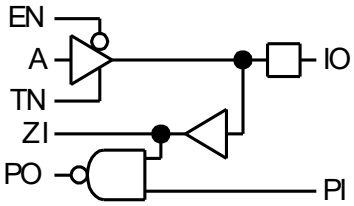
Internal Macrocells			
Name	**Logic Symbol**	**Netlist Notation**	**Function**
AN2, AN2P		Gate_Name(Z) = AN2(A,B)	AND2
AN3, AN3P		Gate_Name(Z) = AN3(A,B,C)	AND3
AN4, AN4P		Gate_Name(Z) = AN4(A,B,C,D)	AND4
AO1, AO1P		Gate_Name(Z) = AO1(A,B,C,D)	AND2 to NOR3
AO2, AO2P		Gate_Name(Z) = AO2(A,B,C,D)	2x AND2 to NOR2
AO3, AO3P		Gate_Name(Z) = AO3(A,B,C,D)	OR2 to NAND3
AO4, AO4P		Gate_Name(Z) = AO4(A,B,C,D)	2x OR2 to NAND2
AO6, AO6P		Gate_Name(Z) = AO6(A,B,C)	AND2 to NOR2
AO7, AO7P		Gate_Name(Z) = AO7(A,B,C)	OR2 to NAND2
B1I		Gate_Name(Z) = B1I(A)	Non inverted Buffer
B2I, B2IP, B3I, B3IP		Gate_Name(Z1,Z2) = B2I(A)	Inverted and non inverted buffer
B4I, B4IP, B5I, B5IP		Gate_Name(Z) = B4I(A)	Inverted buffer
D24L		Gate_Name(Z0,Z1,Z2,Z3) = D24L(A,B)	2 to 4 decoder if BA=HH, Z3=L, if BA=HL, Z2=L, if BA=LH, Z1=L, if BA=LL, Z0=L
EN, ENP		Gate_Name(Z) = EN(A,B)	EXNOR2

EO, EOP		Gate_Name(Z) = EO(A,B)	EXOR2
EO1, EO1P		Gate_Name(Z) = EO1(A,B,C,D)	AND2 and NOR2 to NOR2
EON1, EON1P		Gate_Name(Z) = EON1(A,B,C,D)	OR2 and NAND2 to NAND2
FA1, FA1P, FA1A		Gate_Name(S,CO) = FA1(CI,A,B)	Full Adder
FD1, FD1P, YFD1, YFD1P		Gate_Name(Q,QN) = FD1(D,CK)	D F/F
FD1S, FD1SP		Gate_Name(Q,QN) = FD1S(D,CK,TI,TE)	D F/F with scan if TE=0, D to Q/QN if TE=1, TI to Q/QN
FD2, FD2P YFD2, YFD2P		Gate_Name(Q,QN) = FD2(D,CK,CL)	D F/F with clear
FD2S		Gate_Name(Q,QN) = FD2S(D,CK,CL,TI,TE)	D F/F with clear and scan if TE=0, D to Q/QN if TE=1, TI to Q/QN
FD3, FD3P, YFD3, YFD3P		Gate_Name(Q,QN) = FD3(D,CK,CL,PR)	D F/F with clear and preset
FD4, FD4P, YFD4, YFD4P		Gate_Name(Q,QN) = FD4(D,CK,PR)	D F/F with preset
FJK1S		Gate_Name(Q,QN) = FJK1S(J,K,CK,TI,TE)	JK F/F with scan if TE=0, JK to Q/QN if TE=1, TI to Q/QN
FJK2, FJK2P		Gate_Name(Q,QN) = FJK2(J,K,CK,CL)	JK F/F with clear

FJK2S		Gate_Name(Q,QN) = FJK2S(J,K,CK,CL,TI,TE)	JK F/F with clear and scan if TE=0, JK to Q/QN if TE=1, TI to Q/QN
FJK3, FJK3P		Gate_Name(Q,QN) = FJK3(J,K,CK,CL,PR)	JK F/F with clear and preset
FT2, FT2P		Gate_Name(Q,QN) = FT2(CK,CL)	T F/F with clear if CK L to H, toggle Q,QN
HA1		Gate_Name(S,CO) = HA1(A,B)	Half Adder
IV, IVP		Gate_Name(Z) = IV(A)	Inverter
IVA, IVAP		Gate_Name(Z) = IVA(A)	Inverter
IVDA		Gate_Name(Y,Z) = IVDA(A)	Inverter to inverter
LD2, LD2P, YLD2, YLD2P		Gate_Name(Q,QN) = LD2(D,GN)	Latch if GN=L, Q=D if GN=H, Q=Q
LD4, LD4P, YLD4, YLD4P		Gate_Name(Q,QN) = LD4(D,GN,CL)	Latch with clear if GN=L, Q=D if GN=H, Q=Q
MUX21H, MUX21HP		Gate_Name(Z) = MUX21H(A,B,S)	2 to 1 selector if S=H, Z=B, if S=L, Z=A
MUX21L, MUX21LP		Gate_Name(Z) = MUX21L(A,B,S)	Inverting 2 to 1 selector if S=H, Z=/B, if S=L, Z=/A
MUX41, MUX41P		Gate_Name(Z) = MUX41(D0,D1,D2,D3,A,B)	4 to 1 selector if BA=HH, Z=D3, if BA=HL, Z=D2, if BA=LH, Z=D1, if BA=LL, Z=D0

MUX81, MUX81P		Gate_Name(Z) = MUX41(D0,D1,D2,D3,D4, D5,D6,D7,A,B,C)	8 to 1 selector if CBA=HHH, Z=D7, if CBA=HHL, Z=D6, if CBA=HLH, Z=D5, if CBA=HLL, Z=D4, if CBA=LHH, Z=D3, if CBA=LHL, Z=D2, if CBA=LLH, Z=D1, if CBA=LLL, Z=D0
ND2, ND2P		Gate_Name(Z) = ND2(A,B)	NAND2
ND3, ND3P		Gate_Name(Z) = ND3(A,B,C)	NAND3
ND4, ND4P		Gate_Name(Z) = ND4(A,B,C,D)	NAND4
ND5, ND5P		Gate_Name(Z) = ND5(A,B,C,D,E)	NAND5
ND6, ND6P		Gate_Name(Z) = ND6(A,B,C,D,E,F)	NAND6
ND8, ND8P		Gate_Name(Z) = ND8(A,B,C,D,E,F,G,H)	NAND8
NR2, NR2P		Gate_Name(Z) = NR2(A,B)	NOR2
NR3, NR3P		Gate_Name(Z) = NR3(A,B,C)	NOR3
NR4, NR4P		Gate_Name(Z) = NR4(A,B,C,D)	NOR4
NR5, NR5P		Gate_Name(Z) = NR5(A,B,C,D,E)	NOR5
NR6, NR6P		Gate_Name(Z) = NR6(A,B,C,D,E,F)	NOR6

NR8, NR8P		Gate_Name(Z) = NR8(A,B,C,D,E,F,G,H)	NOR8
OR2, OR2P		Gate_Name(Z) = OR2(A,B)	OR2
OR3, OR3P		Gate_Name(Z) = OR3(A,B,C)	OR3
OR4, OR4P		Gate_Name(Z) = OR4(A,B,C,D)	OR4
YD24GH		Gate_Name(Z0,Z1,Z2,Z3) = YD24GH(A,B,GN)	2 to 4 decoder if BA=HH, Z3=H, if BA=HL, Z2=H, if BA=LH, Z1=H, if BA=LL, Z0=H
YLD24B		Gate_Name(QA,QAN,QB,QBN, QC,QCN,QD,QDN) = YD24B(DA,DB,DC,DD,GN)	Quad latch if GN=L, QA=DA, QB=DB, QC=DC, QD=DD if GN=H, QA=QA, QB=QB, QC=QC, QD=QD
YMUX24H, YMUX24HP		Gate_Name(Z1,Z2,Z3,Z4) = YMUX24H(A1,B1,A2,B2,A3,B3, A4,B4,S)	Quad 2 to 1 selector if S=H, Z1=B1, Z2=B2, Z3=B3, Z4=B4 if S=L, Z1=A1, Z2=A2, Z3=A3, Z4=A4
YMUX24L, YMUX24LP		Gate_Name(Z1,Z2,Z3,Z4) = YMUX24H(A1,B1,A2,B2,A3,B3, A4,B4,S)	Quad Inverting 2 to 1 selector if S=H, Z1=/B1, Z2=/B2, Z3=/B3, Z4=/B4 if S=L, Z1=/A1, Z2=/A2, Z3=/A3, Z4=/A4

I/O Macrocells			
Name	**Logic Symbol**	**Netlist Notation**	**Function**
TLCHT		Gate_Name(Z,PO) = TLCHT(A,PI)	Input buffer
BT2		Gate_Name(Z) = BT2(A,EN,TN)	Tri-state output buffer 2mA if EN=L & TN=H, Z=A if EN=H, Z=HiZ if TN=L, Z=HiZ
BT4		Gate_Name(Z) = BT4(A,EN,TN)	Tri-state output buffer 4mA
BT8		Gate_Name(Z) = BT8(A,EN,TN)	Tri-state output buffer 8mA
BD2T		Gate_Name(IO,ZI,PO) = BD2T(A,EN,TN,PI)	Bi-directional output buffer 2mA if EN=L & TN=H, IO=A if EN=H, IO=HiZ if TN=L, IO=HiZ
BD4T		Gate_Name(IO,ZI,PO) = BD4T(A,EN,TN,PI)	Bi-directional output buffer 4mA

Downloadable Files