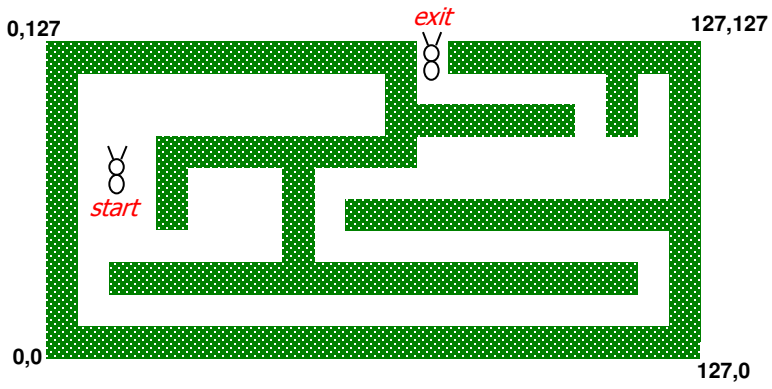
Ant-brain FSM

Run the ant through the maze

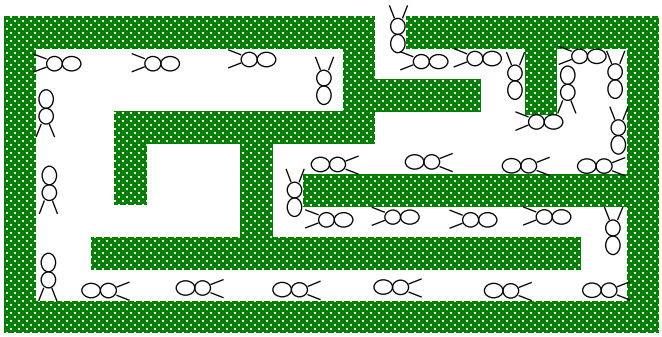
Ant in a maze

- Electronic ant, electronic maze
  - Design the ant



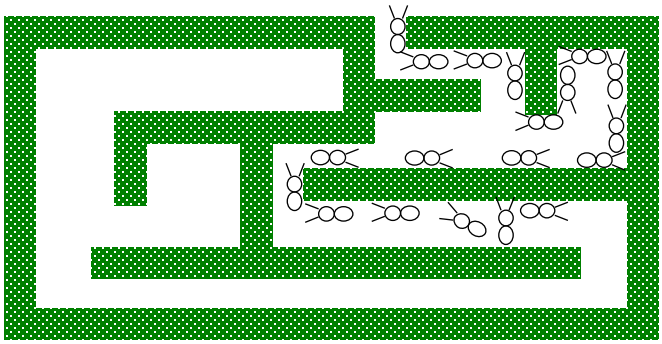
Example: ant brain (Ward, MIT)

- Sensors: L and R antennae, 1 if in touching wall
- Actuators: F - forward step, TL/TR - turn left/right slightly
- Goal: find way out of maze
- Strategy: keep the wall on the right



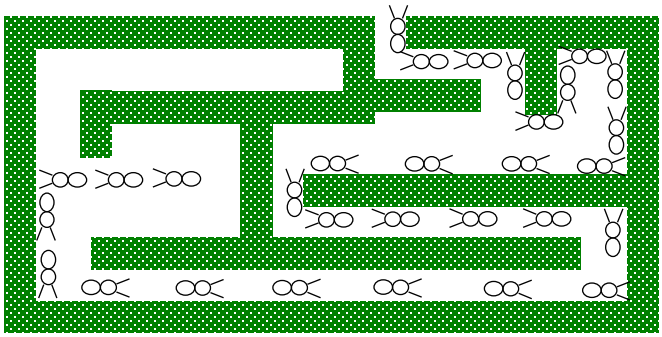
Example: ant brain (special case 1)

- Left (L) Antenna touching the wall



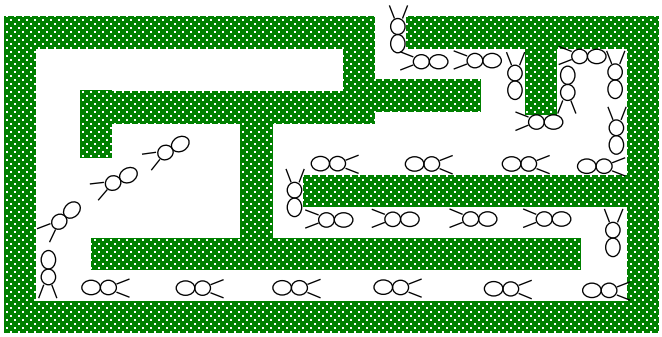
Example: ant brain (special case 2)

- Ant Lost

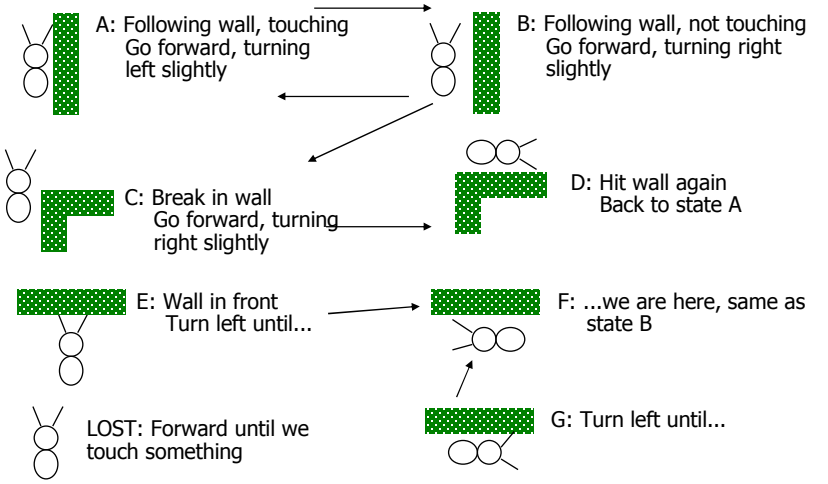


Example: ant brain (special case 2)

- Ant Lost (another example



Ant behavior



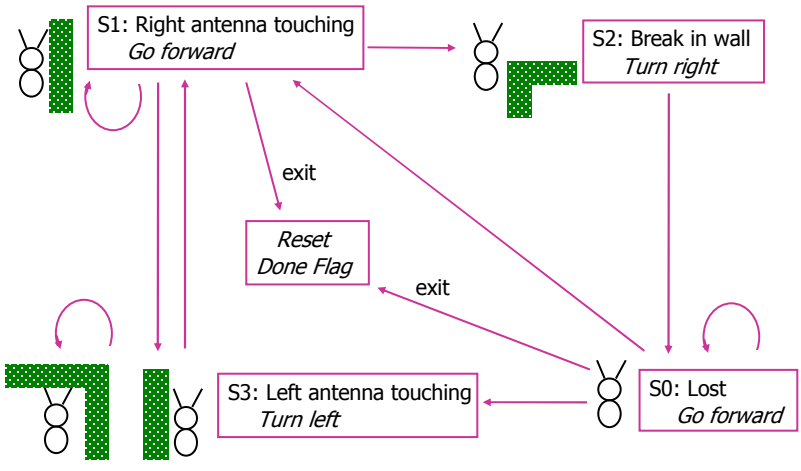
Goal: Find a way out of maze

- Sensors on L and R antennae
  - Sensor = "1" if touching wall; "0" if not touching wall
    - L'R' ≡ no wall
    - L'R ≡ wall on right
    - LR' ≡ wall on left
    - LR ≡ wall in front
    - *** ≡ exit
- Movement:
  - F ≡ forward one step
  - TL ≡ turn left 90 degrees
  - TR ≡ turn right 90 degrees

Notes & strategy

- Notes
  - Maze has no islands
  - Corridors are wider than ant
  - Don't worry about startup
  - Assume a Moore machine
  - Assume D flip-flops
- Strategy
  - Partition your design into datapath and control
  - Keep the wall on the right

The ant's behavior



The maze

- Virtual maze
  - 128 × 128 grid
    - Stored in memory
    - 16384 8-bit words
  - *YX* is maze addresses
    - *X* is the ant's horizontal position (7 bits)
    - *Y* is the ant's vertical position (7 bits)
  - Each memory location says
    - 00000001 ≡ No wall
    - 00000010 ≡ North wall
    - 00000100 ≡ West wall
    - 00001000 ≡ South wall
    - 00010000 ≡ East wall
    - 00100000 ≡ Exit

Can have multiple walls
Example: 00001100
⇒ Walls on South and East

Where do you start???

Don't look ahead

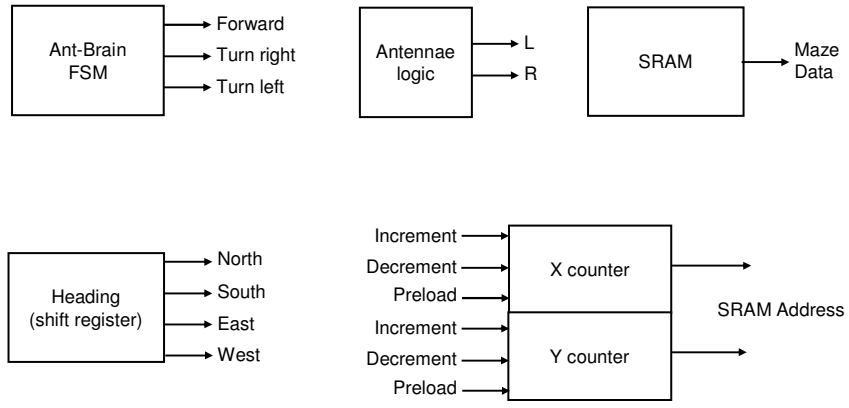
What you need

- An FSM for the ant
  - 3 outputs
    - Go forward
    - Turn left
    - Turn right
- Two 7-bit registers for *X* and *Y*
  - With preload, increment, decrement
- A register to hold the ant's heading
- Logic to convert memory data to antennae info

Recommendations

- 7-bit counters for *X*, *Y*
  - Move horizontally: Increment or decrement *X*
  - Move vertically: Increment or decrement *Y*
- Shift register for heading
  - N: 0001
  - W: 0010
  - S: 0100
  - E: 1000
  - Rotate right when ant turns right
  - Rotate left when ant turns left
- Combinational logic for antennae decoder

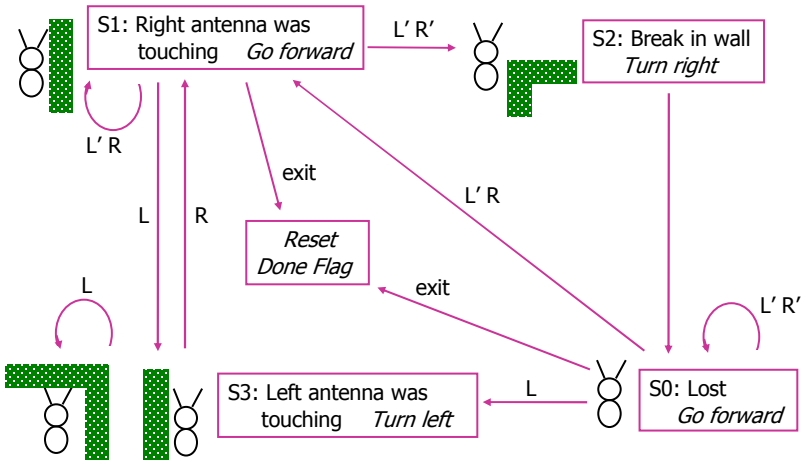
Partition the design



Design the ant-brain FSM

1. State diagram and state-transition table
2. State minimization
3. State assignment (or state encoding)
4. Minimize next-state logic
5. Implement the design

Step 1a: State diagram



Step 1b: State-transition table

Exit	State	L	R	Next State	Output
1	Reset				
0	S0	0	0	S0	F
		0	1	S1	F
		1	0	S3	F
		1	1	S3	F
0	S1	0	0	S2	F
		0	1	S1	F
		1	0	S3	F
		1	1	S3	F
0	S2	0	0	S0	TR
		0	1	S0	TR
		1	0	S0	TR
		1	1	S0	TR
0	S3	0	0	S1	TL
		0	1	S1	TL
		1	0	S3	TL
		1	1	S3	TL

Step 2: State minimization

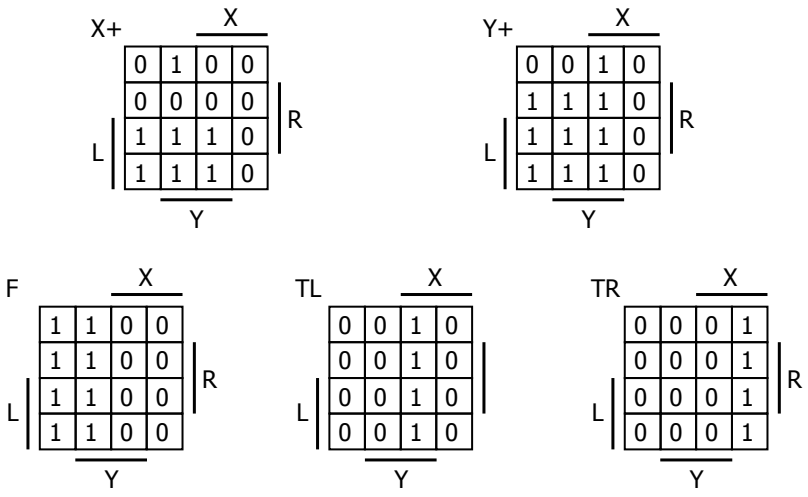
- Two states are equivalent if they cannot be distinguished at the outputs of the FSM
  - The outputs are the same for any input sequence
- Two conditions for two states to be equivalent
  1) Outputs must be the same in both states
  2) Machine must transition to equivalent states for all inputs
- Any equivalent states in our state diagram?

Step 3: State encoding

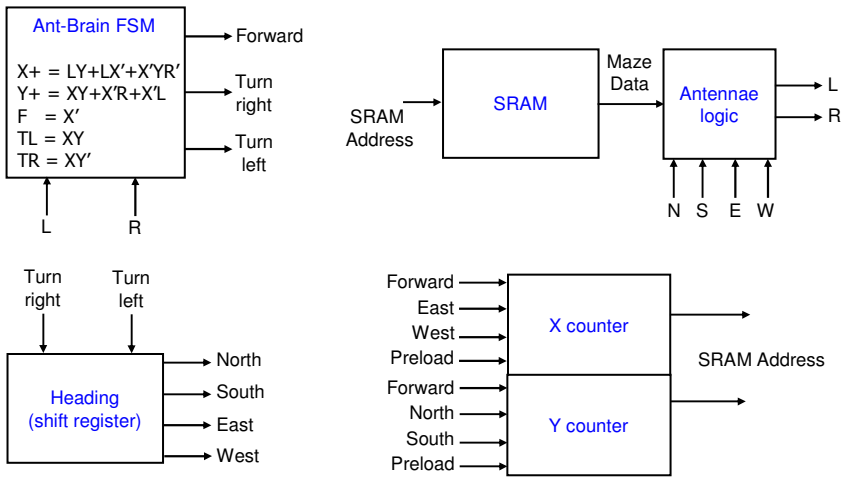
Exit	X Y	L R	X^+ Y^+	F TL TR
1	Reset			
0	0 0	0 0	0 0	1 0 0
	0 0	0 1	0 1	1 0 0
	0 0	1 0	1 1	1 0 0
	0 0	1 1	1 1	1 0 0
0	0 1	0 0	1 0	1 0 0
	0 1	0 1	0 1	1 0 0
	0 1	1 0	1 1	1 0 0
	0 1	1 1	1 1	1 0 0
0	1 0	0 0	0 0	0 0 1
	1 0	0 1	0 0	0 0 1
	1 0	1 0	0 0	0 0 1
	1 0	1 1	0 0	0 0 1
0	1 1	0 0	0 1	0 1 0
	1 1	0 1	0 1	0 1 0
	1 1	1 0	1 1	0 1 0
	1 1	1 1	1 1	0 1 0

S0 ↗ 00
S1 ↗ 00
S2 ↗ 10
S3 ↗ 11

Step 4: Minimize the logic



Step 5: Implement the design



Antennae logic

- Each memory location says
  - 00000001 ≡ No wall
  - 00000010 ≡ North wall (NW)
  - 00000100 ≡ West wall (WW)
  - 00001000 ≡ South wall (SW)
  - 00010000 ≡ East wall (EW)
  - 00100000 ≡ Exit
- The ant can be heading
  - N: 0001
  - W: 0010
  - S: 0100
  - E: 1000

Logic for right antennae

R = NW(N + W) +
WW(W + S) +
SW(S + E) +
EW(E + N)

Logic for left antennae

L = NW(N + E) +
WW(W + N) +
SW(S + W) +
EW(E + S)

Gate count:
4 2-input ORs
8 2-input ANDs
2 4-input ORs

What we left out...

- Crumbs in cell
  - Ant eats crumbs in every cell it visits
  - Writes crumb file back to SRAM
  - Read crumb file, for future display on monitor
- Need a memory controller
  - A state machine to talk to the SRAM
- Need to deal with startup, exit states
- Homework #7, due March 5:

Produce a completed ant-brain design, with documentation, including the above points on this page. SHOW YOUR CHANGES!

Winbond Electronics Corp.

32K × 8 HIGH SPEED CMOS STATIC RAM

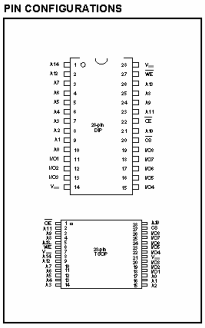
GENERAL DESCRIPTION

The W24257A is a high speed, low power CMOS static RAM organized as 32768 × 8 bits that operates on a single 5-volt power supply. This device is manufactured using Winbond's high performance CMOS technology.

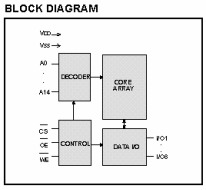
FEATURES

- High speed access time: 10/12/15/20 nS (max.)
- Low power consumption:
  - Active: 400 mW (typ.)
- Single +5V power supply
- Fully static operation
- All inputs and outputs directly TTL compatible
- Three-state outputs
- Available packages: 28-pin 300 mil SOJ, 330 mil SOP, skinny DIP and standard type one TSOP (8 mm × 13.4 mm)

PIN CONFIGURATIONS



BLOCK DIAGRAM



PIN DESCRIPTION

SYMBOL	DESCRIPTION
A0–A14	Address Inputs
I/O1–I/O8	Data Inputs/Outputs
$\overline{CS}$	Chip Select Input
$\overline{WE}$	Write Enable Input
$\overline{OE}$	Output Enable Input
VDD	Power Supply
VSS	Ground

Winbond Electronics Corp.

DC CHARACTERISTICS

Absolute Maximum Ratings

PARAMETER	RATING	UNIT
Supply Voltage to VSS Potential	-0.5 to +7.0	V
Input/Output to VSS Potential	-0.5 to VDD +0.5	V
Allowable Power Dissipation	1.0	W
Storage Temperature	-65 to +150	°C
Operating Temperature	0 to +70	°C

Note: Exposure to conditions beyond those listed under Absolute Maximum Ratings may adversely affect the life and reliability of the device.

TRUTH TABLE

$\overline{CS}$	$\overline{OE}$	$\overline{WE}$	MODE	I/O1–I/O8	VDD CURRENT
H	X	X	Not Selected	High Z	ISB, ISB1
L	H	H	Output Disable	High Z	IDD
L	L	H	Read	Data Out	IDD
L	X	L	Write	Data In	IDD

OPERATING CHARACTERISTICS

(VDD = 5V ± 10%, VSS = 0V, TA = 0 to 70° C)

PARAMETER	SYM.	TEST CONDITIONS		MIN.	TYP.	MAX.	UNIT
Input Low Voltage	VIL	-		-0.5	-	+0.8	V
Input High Voltage	VIH	-		+2.2	-	VDD +0.5	V
Input Leakage Current	ILI	VIN = VSS to VDD		-10	-	+10	µA
Output Leakage Current	ILO	VI/O = VSS to VDD, $\overline{CS}$ = VIH or $\overline{OE}$ = VIH or $\overline{WE}$ = VIL		-10	-	+10	µA
Output Low Voltage	VOL	IOL = +8.0 mA		-	-	0.4	V
Output High Voltage	VOH	IOH = -4.0 mA		2.4	-	-	V
Operating Power Supply Current	IDD	$\overline{CS}$ = VIL, I/O = 0 mA Cycle = MIN Duty = 100%	10	-	-	170	mA
			12	-	-	160	mA
			15			150	mA
			20	-	-	140	mA
Standby Power Supply Current	ISB	$\overline{CS}$ = VIH Cycle = MIN, Duty = 100%		-	-	30	mA
	ISB1	$\overline{CS}$ ≥ VDD -0.2V		-	-	10	mA

Note: Typical characteristics are at VDD = 5V, TA = 25° C.

Winbond Electronics Corp.

CAPACITANCE

(VDD = 5V, TA = 25° C, f = 1 MHz)

PARAMETER	SYM.	CONDITIONS	MAX.	UNIT
Input Capacitance	CIN	VIN = 0V	8	pF
Input/Output Capacitance	CI/O	VOUT = 0V	10	pF

Note: These parameters are sampled but not 100% tested.

THERMAL RESISTANCE

PARAMETER	SYM.	CONDITIONS	MAX.	UNIT
Junction to Case Thermal Resistance	θJC	A. F. R. = 1m/sec, TA = 25° C	20	°C/W
Junction to Ambient Thermal Resistance	θJA	A. F. R. = 1m/sec, TA = 25° C	60	°C/W

Note: These parameters are only applied to "TSOP" and "SOJ" package types.

AC TEST CONDITIONS

PARAMETER	CONDITIONS
Input Pulse Levels	0V to 3V
Input Rise and Fall Times	5 nS
Input and Output Timing Reference Level	1.5V
Output Load	CL = 30 pF, IOH/IOL = -4 mA/8 mA

Winbond Electronics Corp.

AC CHARACTERISTICS

(VDD = 5V ± 10%, VSS = 0V, TA = 0 to 70° C)

Read Cycle

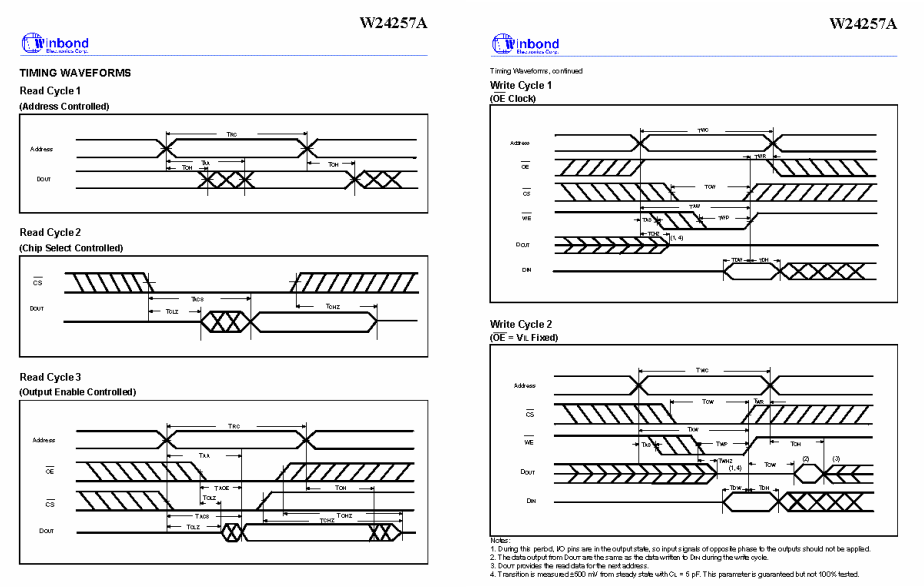
PARAMETER	SYM.	W24257A-10		W24257A-12		W24257A-15		W24257A-20		UNIT
		MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	
Read Cycle Time	TRC	10	-	12	-	15	-	20	-	nS
Address Access Time	TAA	-	10	-	12	-	15	-	20	nS
Chip Select Access Time	TACS	-	10	-	12	-	15	-	20	nS
Output Enable to Output Valid	TAOE	-	5	-	6	-	7	-	10	nS
Chip Selection to Output in Low-Z	TCLZ*	3	-	3	-	3	-	3	-	nS
Output Enable to Output in Low-Z	TOLZ*	0	-	0	-	0	-	0	-	nS
Chip Deselection to Output in High Z	TCHZ*	-	5	-	6	-	7	-	10	nS
Output Disable to Output in High Z	TOHZ*	-	5	-	6	-	7	-	10	nS
Output Hold from Address Change	TOH	3	-	3	-	3	-	3	-	nS

* These parameters are sampled but not 100% tested.

Write Cycle

PARAMETER	SYM.	W24257A-10		W24257A-12		W24257A-15		W24257A-20		UNIT
		MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	
Write Cycle Time	TWC	10	-	12	-	15	-	20	-	nS
Chip Selection to End of Write	TCW	9	-	10	-	13	-	17	-	nS
Address Valid to End of Write	TAW	9	-	10	-	13	-	17	-	nS
Address Setup Time	TAS	0	-	0	-	0	-	0	-	nS
Write Pulse Width	TWP	9	-	10	-	10	-	12	-	nS
Write Recovery Time ($\overline{CS}$, $\overline{WE}$)	TWR	0	-	0	-	0	-	0	-	nS
Data Valid to End of Write	TDW	6	-	7	-	9	-	10	-	nS
Data Hold from End of Write	TDH	0	-	0	-	0	-	0	-	nS
Write to Output in High Z	TWHZ*	-	6	-	7	-	8	-	10	nS
Output Disable to Output in High Z	TOHZ*	-	6	-	7	-	8	-	10	nS
Output Active from End of Write	TOW	0	-	0	-	0	-	0	-	nS

* These parameters are sampled but not 100% tested.



W24257A

TIMING WAVEFORMS

Read Cycle 1

(Address Controlled)

Read Cycle 2

(Chip Select Controlled)

Read Cycle 3

(Output Enable Controlled)

W24257A

Timing Waveforms, continued

Write Cycle 1

(OE Clock)

Write Cycle 2

(OE = VIL Fixed)

Notes:
1. During this period, I/O pins are in the output state, so input signals of opposite phase to the outputs should not be applied.
2. The data output from DOUT are the same as the data written to DIN during the write cycle.
3. DOUT provides the read data for the next address.
4. Transition is measured ±500 mV from steady state with CL = 5 pF. This parameter is guaranteed but not 100% tested.

Memory initialization

```
Module sram(xx, xx, xx, xx,…xx)
   …

initial
   $readmemb("mazefilename",arrayname,start,length);
…
…
…
   begin
        file = $fopen("dumpfilename"0;
        for (……)
          $fdisplay(file,arrayname[index];
        $fclose(file);
   end
endmodule
```