Projects Files up
- Use chipscope
  "Trigger on data" - select window

DRAM
  Asynch
  Synch

Project

DRAM last time
  Address
  RAS#
  CAS#
  WE#
  CS#
  Data

Output enable

t_{RP}

RAS -> CAS delay read out from memory cell array
  CAS latency t_{CL}
  CAS# to data valid
  Row precharge t_{RP}
  RAS# write back next row

Writing
  Row Reg

Read Row
  Modify
  Write back

Interleaved reads/writes OK

Refresh needed

2^n * t_{RC} every

64 m s

Auto, or external

Row cycle

(without only 1 read)
why is tCL so long? 16-44 tCL
what if we group them by 4?
256 groups of 4
pipeline de-mux

SDRAM - add a clock to keep things straight
CS
WE
RAS
CAS

Inputs to state machine.

Separate state machine for each bank
(mostly)
RAS
CAS
WE

Activate
L
H
L
Read
L
H
H
Write
H
L
H
Pre-charge
Refresh, Self refresh (timed, all rows)

one row

access is so slow, why not share pins w/
parallel hadware?
4, 8 banks
2, 3 bit bank address
4, 8 copies of everything on chip.

BA
2
enable lines
RAS
CAS
RAS
CAS
RAS
CAS

RAS
CAS

IDLE
Activate
(tRED)

WRITE
PRE
READ

READ
PRE

WRITE

RAS
CAS

READ

PRE

RAS
CAS

RAS
CAS

TRP
Our chip 256MB as 32M x 8B = 32M x 64b
4 chips, each 32M x 16b = 512MB
each chip has 4 banks of 128MB

The bus runs @ 200MHz (5ns)

$t_{RCD} = 15\, \text{ns} \quad (3)$
$t_{CL} = 15\, \text{ns} \quad (3)$
$t_{RP} = 15\, \text{ns} \quad (3)$
$t_{RC} = 55\, \text{ns} \quad (11)$

Bank - bank activate delay: 10ns
burst length = 4

Each bank has 8,194 rows
of 16,384 bits

Read a burst

CMD $\frac{AC}{R}$ READ
Addr rowX colX
Data

Latency = 30ns

Random access BW

\[
\frac{1\, \text{fetch}}{55\, \text{ns}} = \frac{18\, \text{M fetch}}{\text{sec.}}
\]

256B = 32B
\[
\frac{10\, \text{ns}}{10\, \text{ns}} = 3.2 \, \text{Gbps}
\]