2BA4 Combined Done and Error Signaling

- Want flexible, economical solution.
- Don't wand to decode unused address space.
- Assume we have ***n*** devices:
  - -> ***n*** chip-selects ($CS_1 \ldots CS_n$)

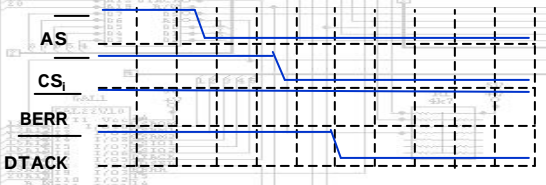
2BA4 Key Idea

- Use common circuit to generate Done-Indicator and Bus-Error signals.
- Want to delay BERR-BAR generation to allow other logic to work.
  - Timing required!

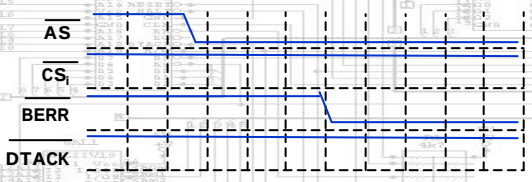
2BA4 Bus –Error Signal

- Assign clock cycle count (CCC_i) to each CS_i, chosen to provide correct number of wait-states for the associated device.
  - CS_1 (ROM) 3 wait states $CCC_1 = 4$
  - CS_2 (RAM) 1 wait state $CCC_2 = 2$
  - CS_3 (RAM) 1 wait state $CCC_3 = 2$

2BA4 CASE 1 –Valid Device



2BA4 CASE 2 – No Valid Device



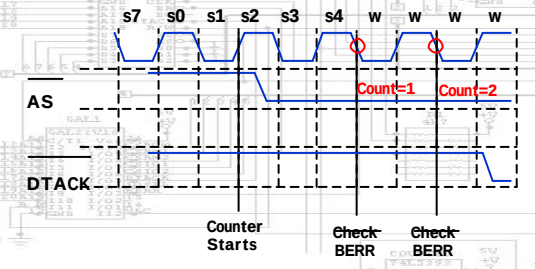
BERR is asserted if DTACK isn't for a time interval after AS was, asserted.

2BA4 Bus Error

- Use CLK as timing signal.
- Count from 0 when AS is asserted.
- Assert BERR if timeout count is reached.
- BERR is unasserted when AS is de-asserted.

2BA4 Idea – Count Clock Cycles

Once AS is asserted.



Count = number of wait-states + 1.
Counting the edges.

2BA4 Process (1)

1. When AS asserted, start counting CPU clock cycles.
2. If CS_i asserted, but incorrect operation, assert BERR and stop counting.
3. If CS_i asserted and count = CCC_i then stop counting and set DTACK.

2BA4

Process (2)

4. If count reaches TOUT, assert BERR and stop counting.
5. When AS is removed, clear counter, and remove Done-Indicator and Bus-Error.

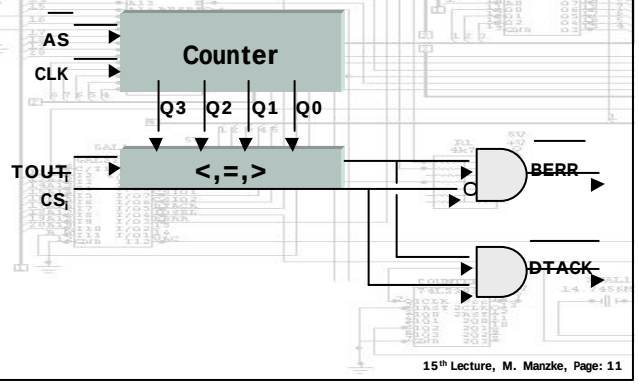
2BA4

e.g. Assume Address selects device$_2$:

1. AS asserted -> count =0
2. Decoding logic asserts CS_2
3. While (count < 5)
   - { Wait}
4. Assert Done-Indicator, stop counting.

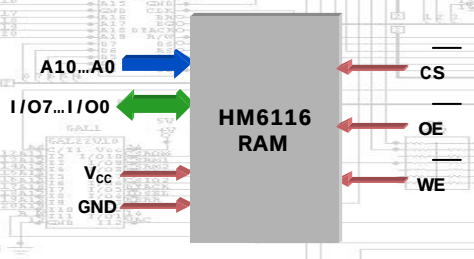
2BA4

Clock Cycle Counter



2BA4

RAM



2BA4 Connecting CPU and RAM

- RAM similar to ROM
- Key difference:
  - Need to use R/$\overline{W}$ line in some way.
- $\overline{CS}$: (Chip Select)
- $\overline{OE}$: (Output Enable) used to read from RAM
- $\overline{WE}$: (Write Enable) used to write to RAM

2BA4 Tutorial (Project Memory Map)

- 20 address bits
- Reset @ $00000
- 1x8k EEPROM
- 2x2k RAM
- 2x4byte I/O devices

2BA4 Project Memory Map

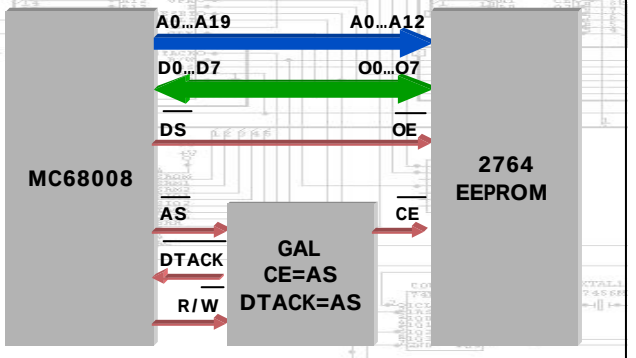
Device	Start	End	Size
ROM_1	$00000	$1FFFF	$2000
RAM_1	$40000	$407FF	$800
RAM_2	$40800	$40FFF	$800
IO_1	$80000	$80003	$4
IO_2	$C0000	$C0003	$4

2BA4 Project Road-Map (1)

- Phase 0: Halt/Reset and Clk circuitry.
- Phase 1: CPU & EEPROM (Cheat).
- Phase 2: + GAL doing “Cheat” wiring.

2BA4

Phase 2: + GAL doing “Cheat” wiring.



2BA4

Project Road-Map (2)

- Phase 3: + GAL doing ROM CE.
- Phase 4: + RAM
- Phase 5: + IO Devices
- Phase 6: + GAL Equations for BERR