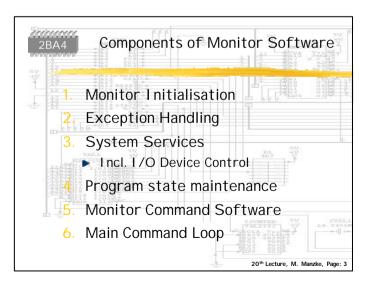
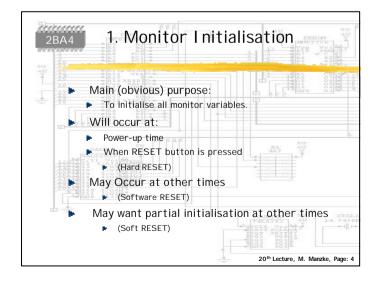
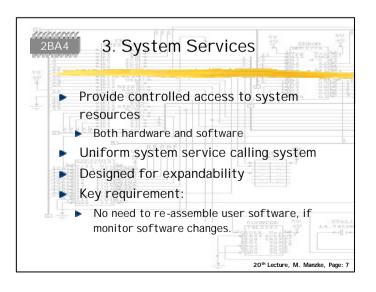
Purpose of the Monitor Software To assist software development by... Host computer communication Loading and running User Software (US) Providing US access to system resources Aiding the debugging Providing a useful interface



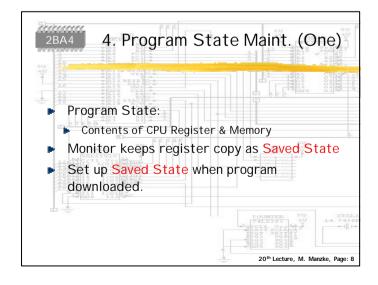
Power-up machine Set up transparent link to host Edit & assemble on host Download software to microprocessor Set breakpoint Start software running At breakpoint: Examine & set register Continue program execution



2. Exception Handling (One) Exceptions must be handled properly Most of little interest or should not occur... Default Exception Handler



2. Exception Handling (Two) I deal Behaviour: Write message saying unexpected exception occurred GIVE DETAIL! Tidy up Return to Monitor Command Loop Needs I/O services Pay careful attention to system stack contents!

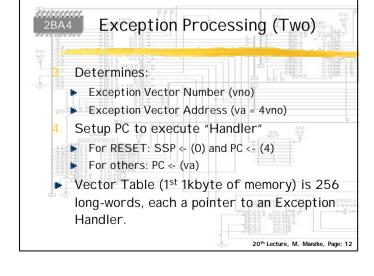


2BA4 4. Program State Maint. (Two) (Re)Start Program: Saved State -> CPU State Stop Program: CPU State -> Saved State Allow user to Query and Modify Saved State. Special care needed with System Stack

and Pointer

When 68008 starts to process an exception it: Saves current (SR, PC) using SSP This is not done for a RESET. Sets new context in Status Register (S=1 [Supervisor Mode], T=0 [Trace off]) Additional information may be pushed onto SSP - depending on exception .

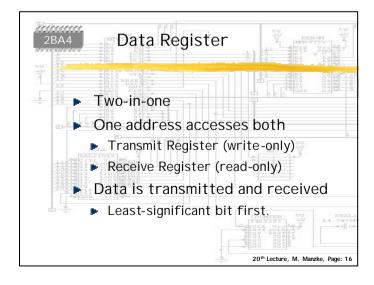
Description of the commands of



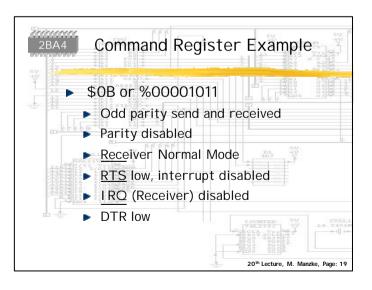
The 68008 divides exceptions into 3 groups Group 0: RESET, Bus Error & Address Error Exception Processing starts immediately. Group 1: Trace, Interrupt, Illegal Instr., Unimplemented Instr., Privilege Violation. Exception Processing starts end end of current instruction. Group 2:Trap Instruction Exception Processing starts as consequence of current instruction

Programming the ACIAs ACIAs take up four bytes Base-Address + 0 ... Base-Address + 3 O: Data Register (Transmit/Receive) Write data for transmission to it. Read received data from it. Status Register Interrupt condition and other status Command Register Controls communication functions 4: Control Register Controls communication speed.

User Programs often want own exception handlers. But Vector Table is fixed in ROM. I dea: Place "real" vector table in RAM (modifiable) Get ROM table to point "through" RAM table Need to initialise RAM table!



Contents set by ACIA A READ transfers contents to CPU A Write causes a programmed RESET of ACIA Bit 7: set if interrupt has occurred Bit 4: set if Transfer Register empty Bit 3: set if Receiver Register full Bit 2: set if overrun has occurred Bit 1: set if framing error has occurred Bit 0: set if parity error occurred



Command Register Contents read/written by CPU Bits 7-6: Parity control Bit 5: Set if Parity mode enabled Bit 3-2: Transfer interrupt control Bit 1: Set if I RQ interrupt enabled 20th Lecture, M. Manzke, Page: 18

