Fundusze Europejskie
Wiedza Edukacja Rozwój

Rzeczpospolita Polska

Unia Europejska
Europejski Fundusz Społeczny

Politechnika Śląska jako Centrum Nowoczesnego Kształcenia
opartego o badania i innowacje

POWR.03.05.00-IP.08-00-PZ1/17

Projekt współfinansowany przez Unię Europejską ze środków Europejskiego Funduszu Społecznego

Microprocessor and Embedded Systems

Faculty of Automatic Control, Electronics and Computer Science, Informatics, Bachelor Degree

Lecture 8

DMA controllers

Bartłomiej Zieliński, PhD, DSc

DMA controllers

Program:

- DMA controller functions
- DMA controllers:
  - Intel 8257 & 8237
  - Zilog Z80 DMA

DMA controllers

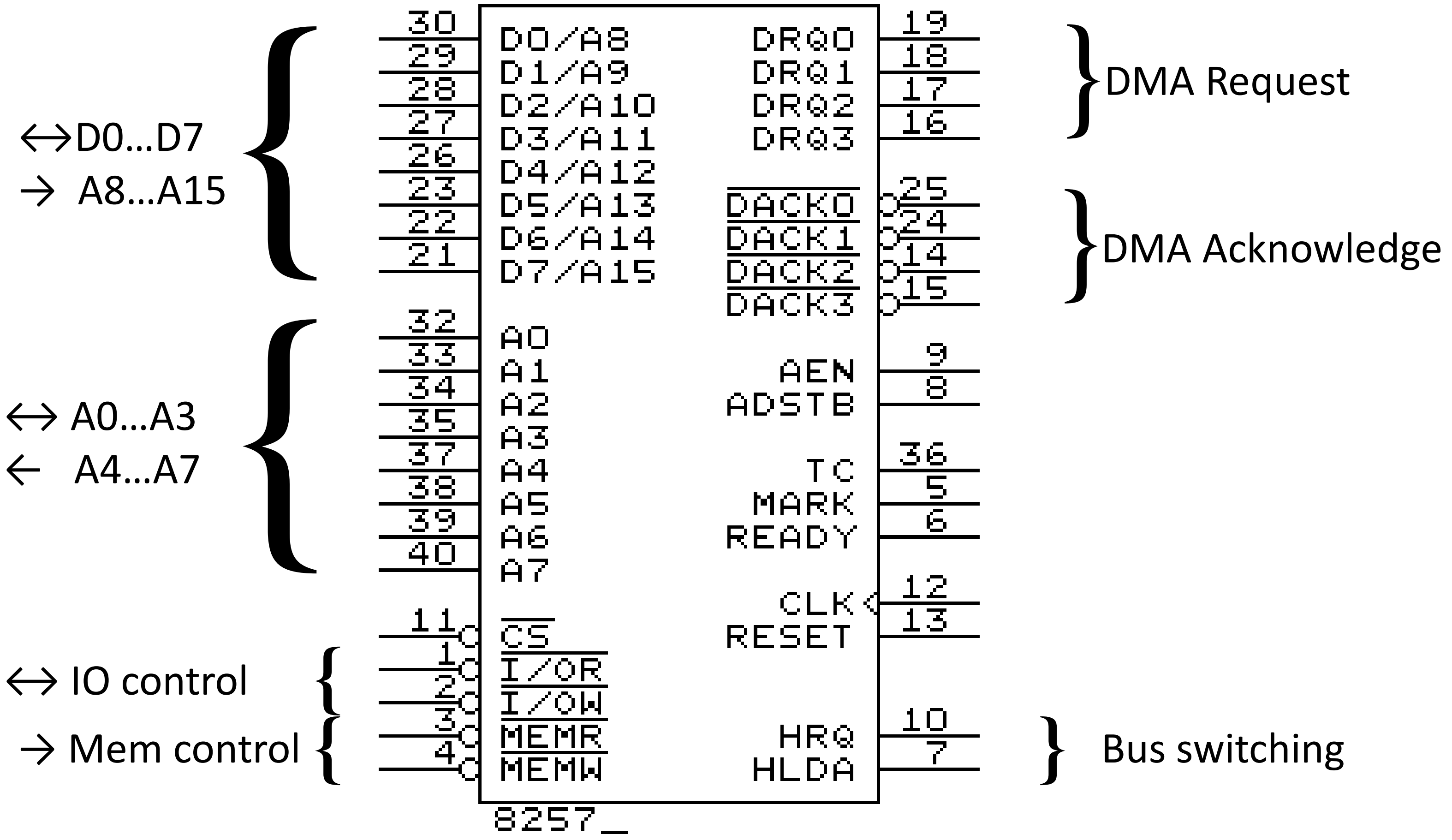
- DMA controller functions
  - **Mem↔IO data transmission on behalf of µp**
    - Bus control
    - DMA source recognition
    - Priority control
    - Mask control
  - Mem↔Mem or IO↔IO transmission
  - Additional data processing during transfer
    - e.g., search for a byte using a pattern
  - Autoinitialisation option

DMA controllers

- Intel 8257
  - 4 independent DMA channels
  - Autoinitialisation option
  - 16-b memory address
  - 14-b block length (up to 16KB)
    - Seems sufficient for 8-bit systems
  - Static or dynamic priority assignment
  - Multiplexed address/data bus

DMA controllers

- Intel 8257 circuit pins



DMA controllers

- 8257 control signals
  - AEN – prevents non-DMA-enabled devices from thinking they're adressed
  - ADSTB – Address Strobe (such as ALE in 8051)
  - TC – Terminal Count – the last byte of the current cycle
  - Mark – a pulse every 128 transferred bytes

DMA controllers

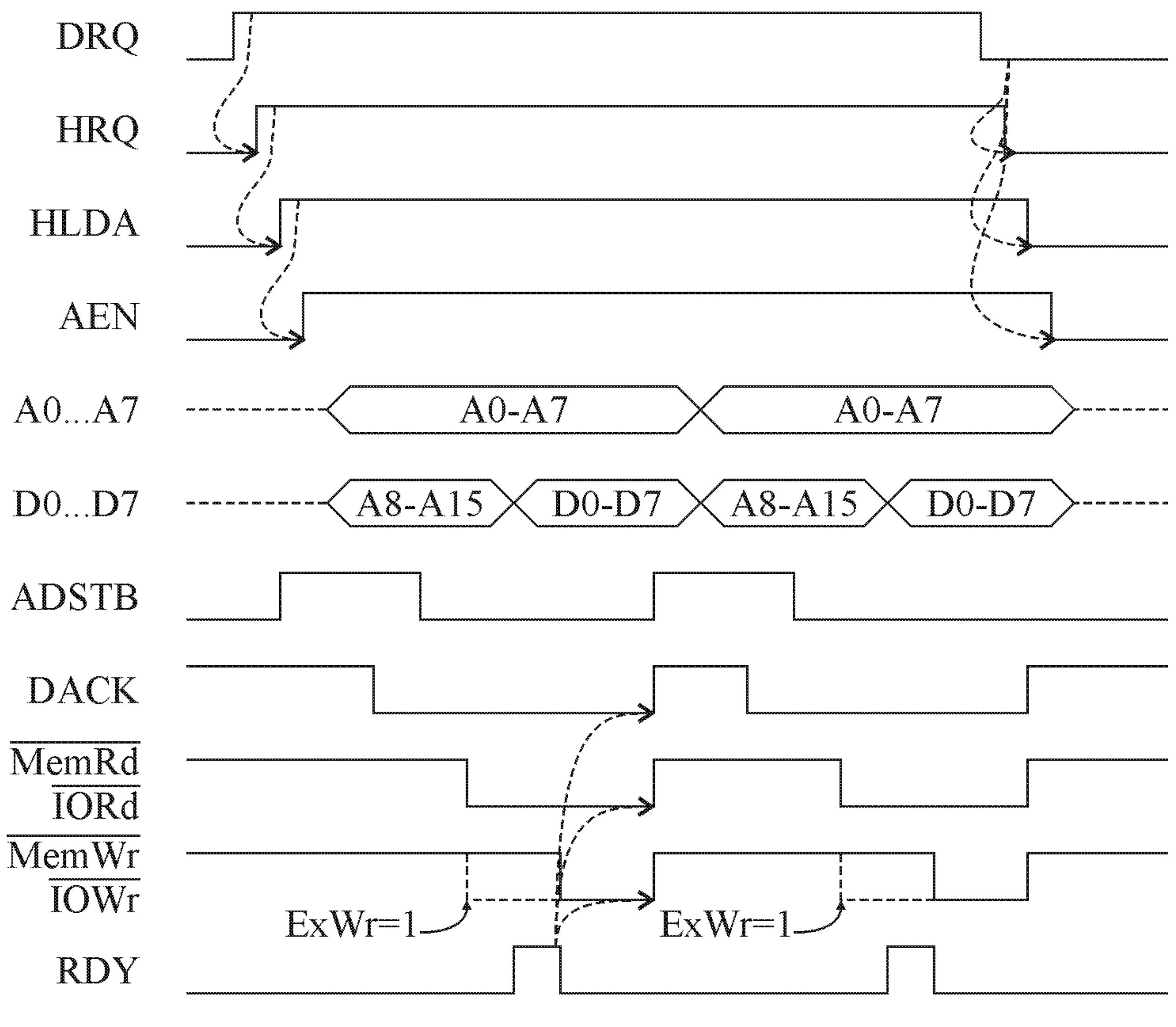
- 8257 operation
  - Passive state
    - 8257 is a typical IO device accessed by µp
    - Address lines are inputs
    - $\overline{\mathrm{IOWr}}, \overline{\mathrm{IORd}}$ lines are inputs
    - $\overline{\mathrm{MemWr}}, \overline{\mathrm{MemRd}}$ unused
  - Active state
    - 8257 takes control over the bus
    - Address lines are outputs
    - $\overline{\mathrm{IOWr}}, \overline{\mathrm{IORd}}$ lines are outputs
    - $\overline{\mathrm{MemWr}}, \overline{\mathrm{MemRd}}$ are outputs

DMA controllers

- DMA transfer
  - Programming DMA controller
    - Mode & options
    - Block length
    - Buffer address
  - Programming IO device
    - E.g., disk controller: which sector to read
  - IO device ready for DMA transfer
  - DMAC takes over bus control
  - Data transfer
  - Bus control returned to µp

DMA controllers

- Intel 8257 data transfer



DMA controllers

- Intel 8257 programming
  - 9 registers
  - 16-b address register (for each channel)
    - LSB/MSB alternate access
  - 16-b length & mode register (for each channel)
    - LSB/MSB alternate access
  - 8-b control register (common)
  - 8-b status register (common)

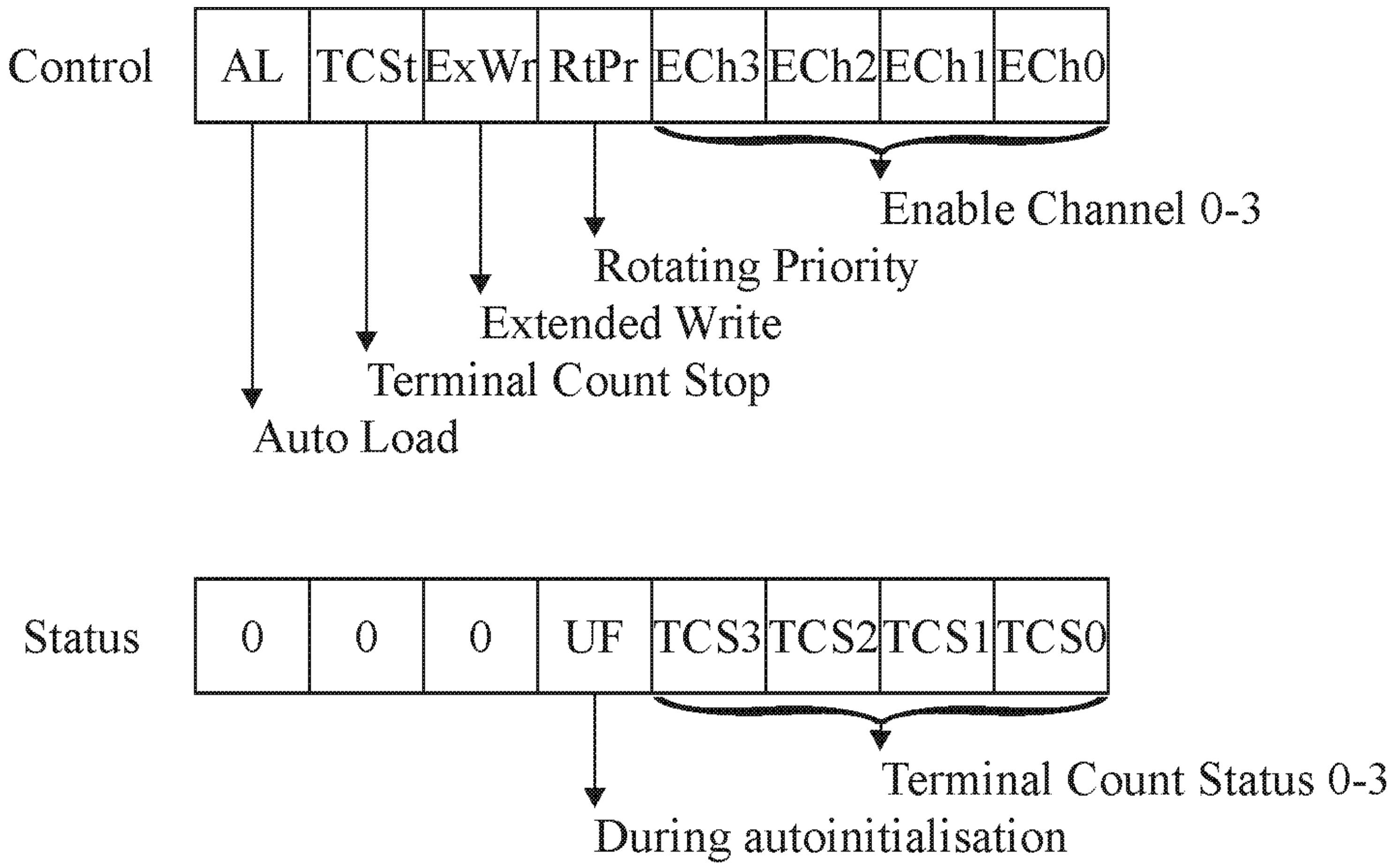
DMA controllers

- Intel 8257 programming

Address (dec)	A3…A0	Register
0	0000	Address #0
1	0001	Counter/mode #0
2	0010	Address #1
3	0011	Counter/mode #1
4	0100	Address #2
5	0101	Counter/mode #2
6	0110	Address #3
7	0111	Counter/mode #3
8	1000	Control/Status

DMA controllers

- Intel 8257 programming



DMA controllers

- Intel 8257 programming
  - Control register:
    - Each channel individually enabled/disabled
    - Rotating priority option
    - Extended Write – $\overline{\mathrm{IOWr}}$, $\overline{\mathrm{MemWr}}$ signal generated earlier to allow RDY be effective
    - Terminal Count Stop – upon Terminal Count, channel stops operation (needs to be programmed again)
    - Auto Load – autoinitialisation
      - Channel 2 – operating
      - Channel 3 – backup only (not usable)
      - Upon TC in ch. 2, it’s programmed with data from ch. 3

DMA controllers

- Intel 8257 programming
  - Status register:
    - Terminal Count information (for each channel individually)
    - UF – during autoinitialisation
      - Prevents from unintentional data loss by reprogramming ch. 3 until its data copied to ch. 2

DMA controllers

- Intel 8257 programming
  - Transfer types
    - Read
      - Mem → IO
    - Write
      - IO → Mem
    - Verification
      - No transfer takes place (no Mem/IO Rd/Wd signals active)
      - Bus control, DMA acknowledge as usual
      - IO device may perform additional internal actions over the data

DMA controllers

- Intel 8237A
  - Pin/signal compatible with 8257
    - Mark output → Pin5 input (=1 or not connected)
    - TC output → $\overline{\mathrm{EOP}}$ i/o (0 → end of process)
  - More functional capabilities
  - More transmission modes
  - More registers
    - Address and length/mode registers are the same

DMA controllers

- Intel 8237A transmission modes
  - Single transfer mode
    - One transfer only
    - DREQ must be active until DACK
    - If DREQ still active, HRQ goes inactive and then active again, etc.
      - DMAC and µp machine cycles interleaving one by one
      - µp can access the bus, if necessary
      - No large delays in bus access by the µp
      - DMA transfer efficiency a bit lower
    - DREQ inactive or TC → end of transfer
      - → autoinitialise (if so programmed)

DMA controllers

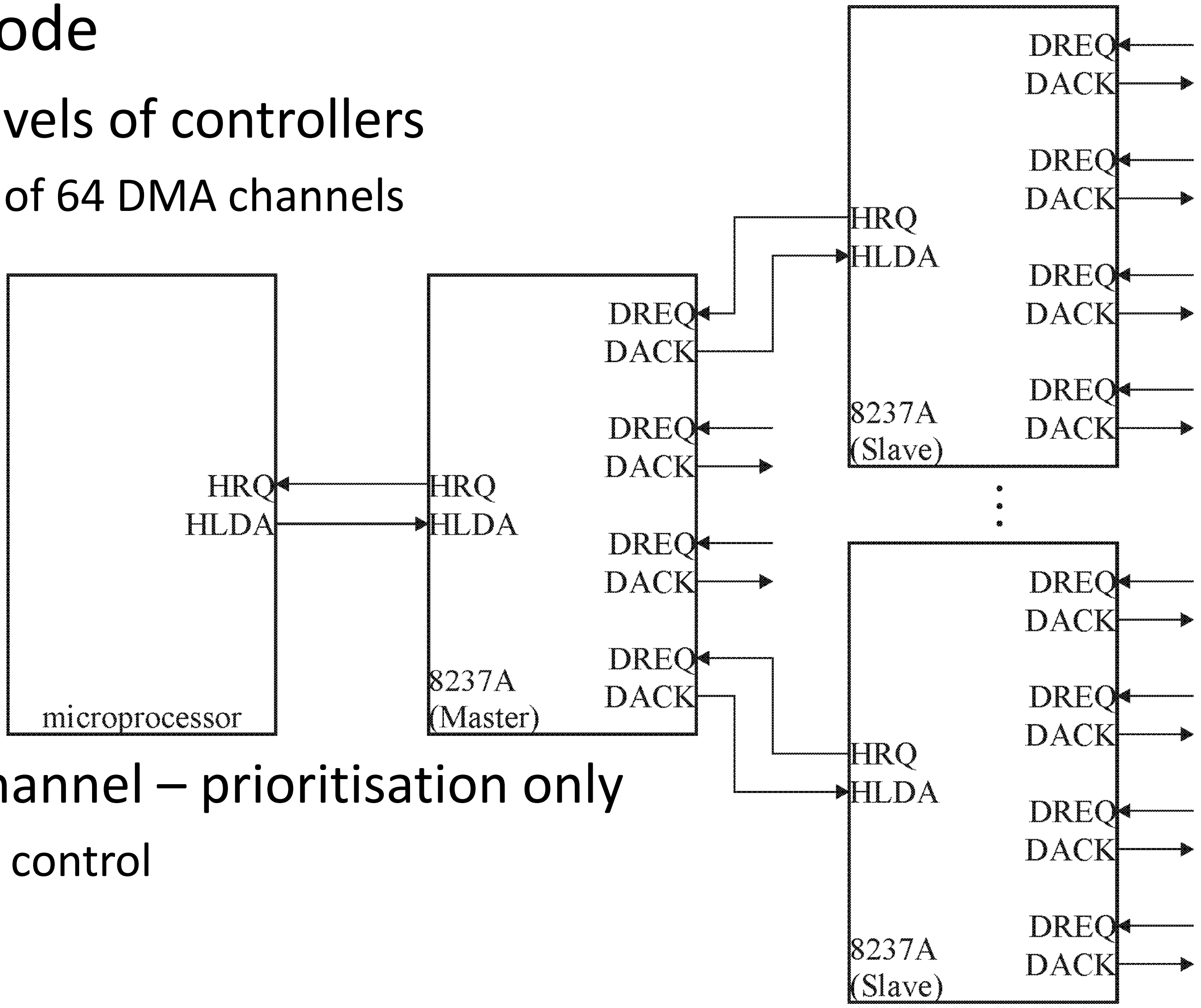
- Intel 8237A transmission modes
  - Block transfer mode
    - Identical transfers repeating
    - DREQ active until DACK
    - After DACK, DREQ state irrelevant
      - DMA transfer lasts until the end
      - µp must wait for bus access
      - DMA transfer efficiency is as high as possible
    - TC or $\overline{\mathrm{EOP}}$ → end of transfer
      - → autoinitialisation (if so programmed)

DMA controllers

- Intel 8237A transmission modes
  - Demand transfer mode
    - Identical transfers repeating
    - DREQ active until DACK
    - After DACK, DREQ held active to continue
      - DMA transfer lasts until the end
      - µp must wait for bus access
      - DMA transfer efficiency is as high as possible
    - DREQ inactive → pause in transfer
      - Register values remembered
    - TC or $\overline{\mathrm{EOP}}$ → end of transfer
      - → autoinitialisation (if so programmed)

DMA controllers

- Intel 8237A transmission modes
  - Cascade mode
    - Up to 3 levels of controllers
      - A total of 64 DMA channels



    - Master channel – prioritisation only
      - No bus control

DMA controllers

- Intel 8237A features
  - Mem→mem transfer
    - Ch. 0 = source, Ch. 1 = destination
    - Start: software DREQ in ch. 0
    - AEN active, DACK inactive
    - Stages:
      - Temp_reg = mem [ch. 0]
        - » Source address: inc, dec or constant
      - Mem [ch. 1] = temp_reg
        - » Destination address: inc or dec
    - $\overline{\mathrm{EOP}}$ → end of transfer

DMA controllers

- Intel 8237A features
  - Autoinitialisation
    - Possible for each channel
    - No cost of channel lost for initialisation data storage
  - Compressed timing
    - A8-A15 updated when needed only
    - Read pulse shorter
    - Less clocks per cycle

DMA controllers

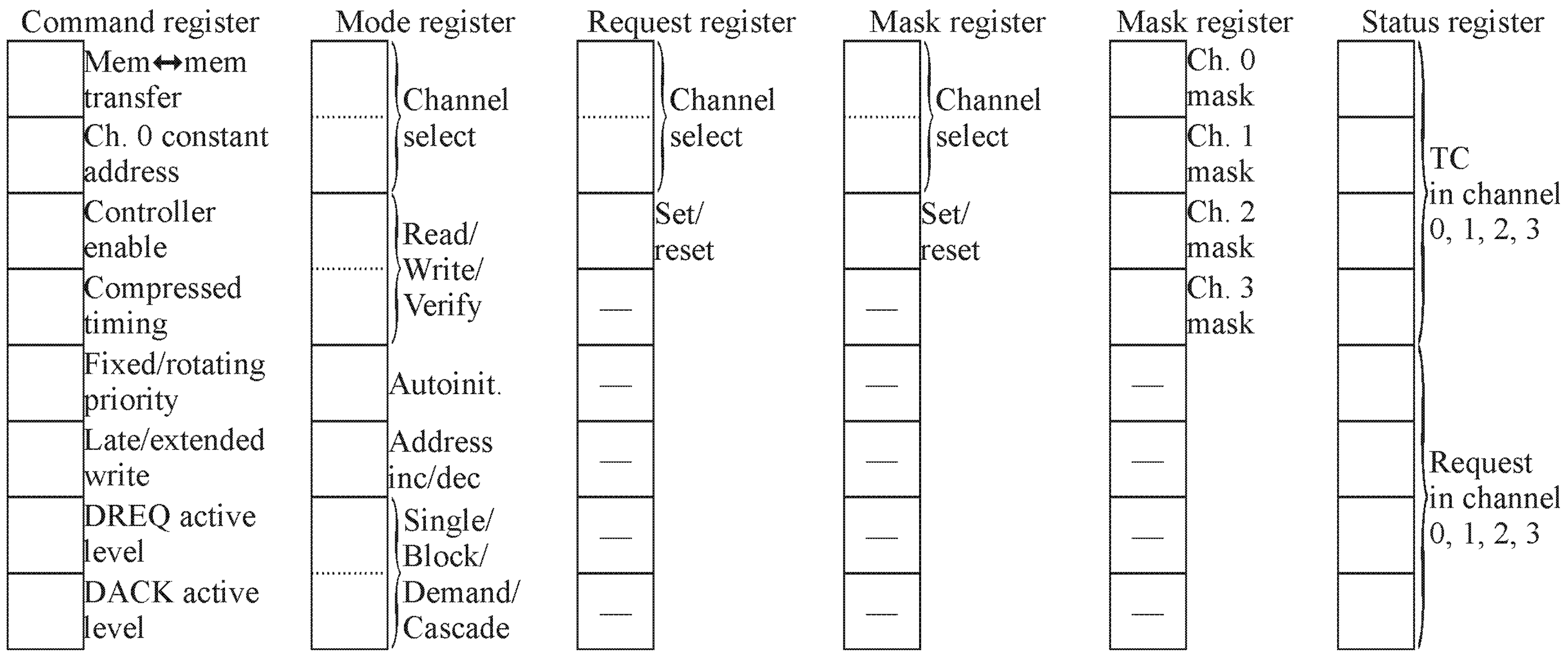
- Intel 8237A programming
  - Control registers

Address (dec)	A3…A0	Register
8	1000	Control (Wr)/Status (Rd)
9	1001	Command (Wr)
10	1010	Mask (set/reset)
11	1011	Mode (Wr)
13	1101	Temp (Rd)
15	1111	Mask (Wr)
12	*1100*	*Clear Byte flip/flop (Wr)*
13	*1101*	*Master Clear (Wr)*
14	*1110*	*Clear Mask Register (Wr)*

} No data needed (address only)

DMA controllers

- Intel 8237A programming

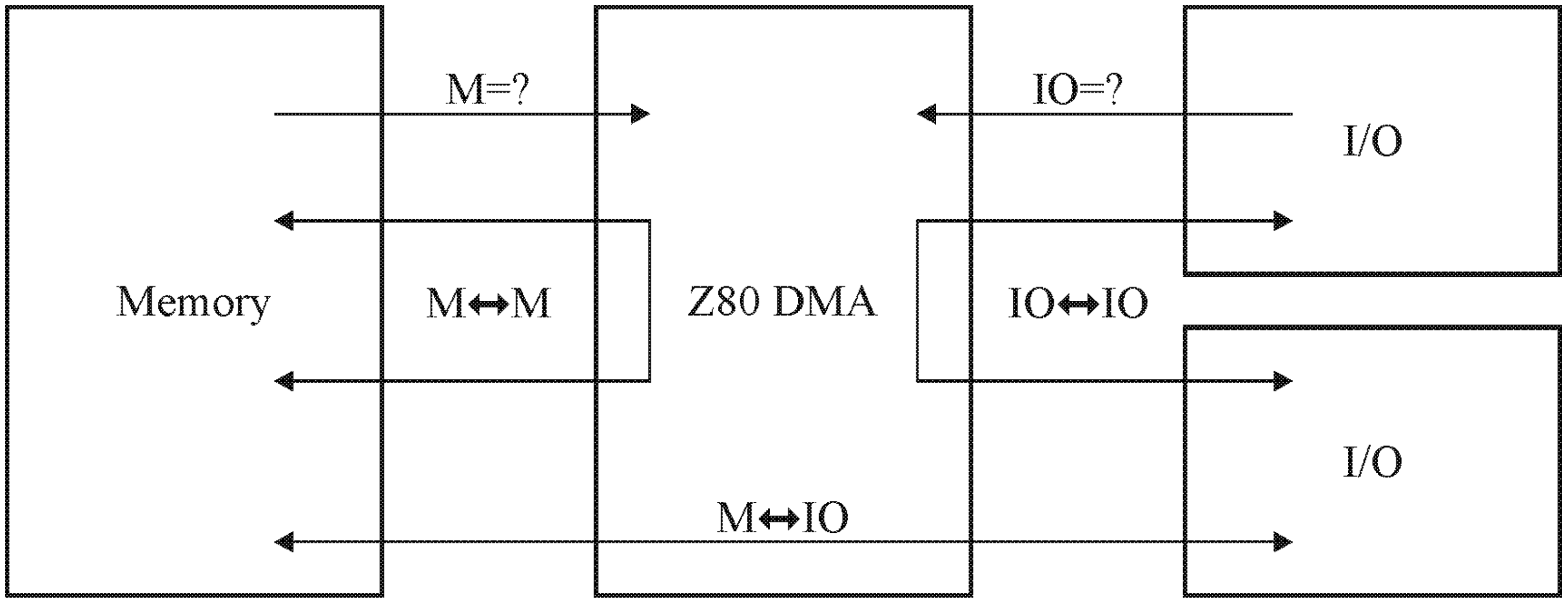


DMA controllers

- Zilog Z80 DMA features
  - 2 DMA channels
  - 16-b address, 8-b data
  - DMA chain & INT chain
  - INT circuit
    - INT conditions
    - INT vector
  - Autoinitialisation possible
  - Programmable „mark” signal (1÷256 bytes)
    - $\overline{\mathrm{INT}}$ used as a „mark”
    - No conflict during DMA cycle

DMA controllers

- Zilog Z80 DMA features
  - Operations:
    - Transmission
    - Comparison
    - Transmission & comparison
    - Mem⟷IO
    - Mem ⟷Mem
    - IO ⟷IO



DMA controllers

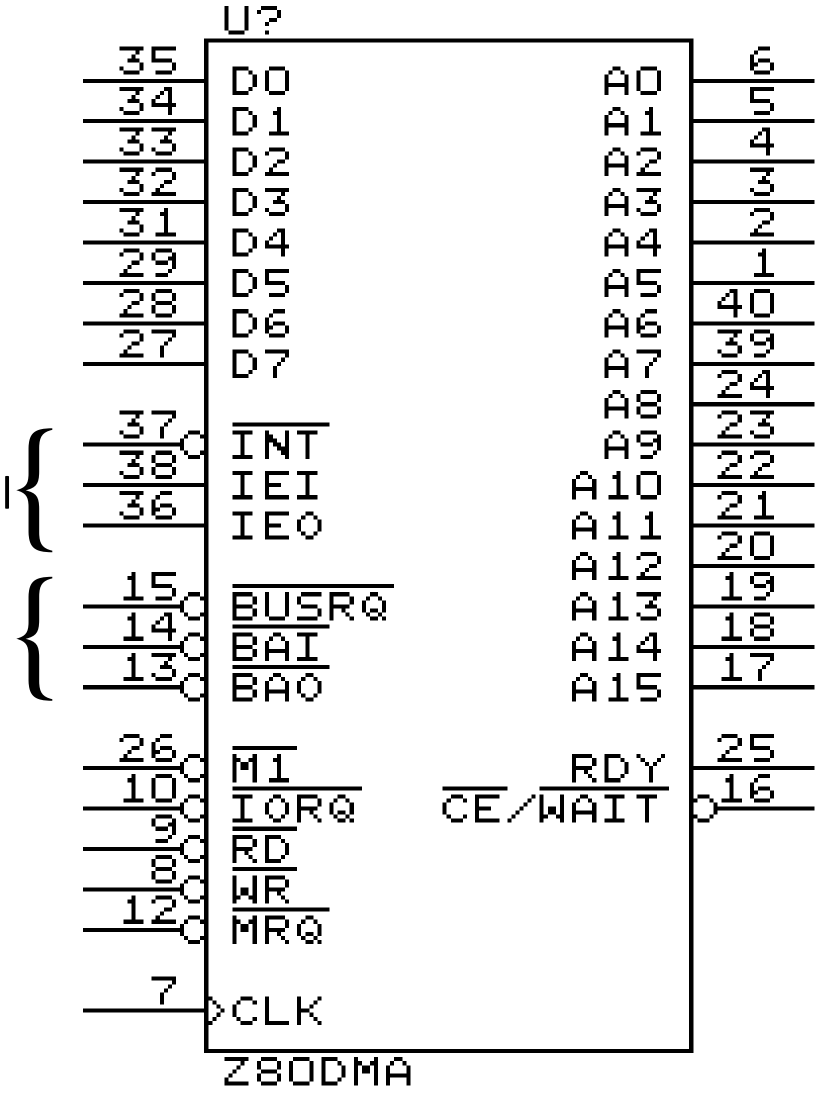
- Zilog Z80 DMA modes
  - Byte mode (single mode, byte-at-a-time mode)
    - One byte at a time
    - Bus released after each byte, and requested again
  - Burst mode (demand mode)
    - Until Ready line is inactive
  - Continuous mode (block mode)
    - Until end of block or data match condition
    - Ready inactive → DMA pause

DMA controllers

- Zilog Z80 DMA circuit pins

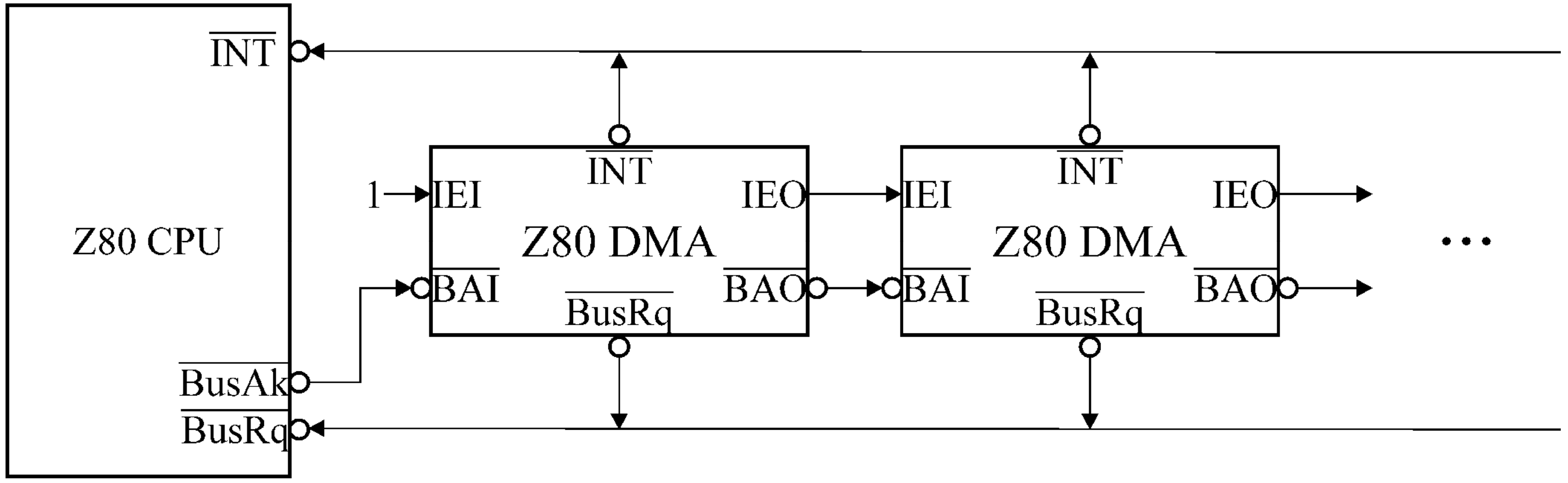
$\leftrightarrow$ int control

$\leftrightarrow$ DMA control



DMA controllers

- Zilog Z80 DMA
  - DMA chain
  - INT chain



DMA controllers

- Zilog Z80 DMA interrupts
  - When RDY active
  - When pattern matched
  - When transfer ends
  - When pattern matched at the block end
- Zilog Z80 DMA programming
  - 7 basic + 14 auxilliary write registers
  - 7 status registers
    - Sequential read in a strict sequence