東邦大学

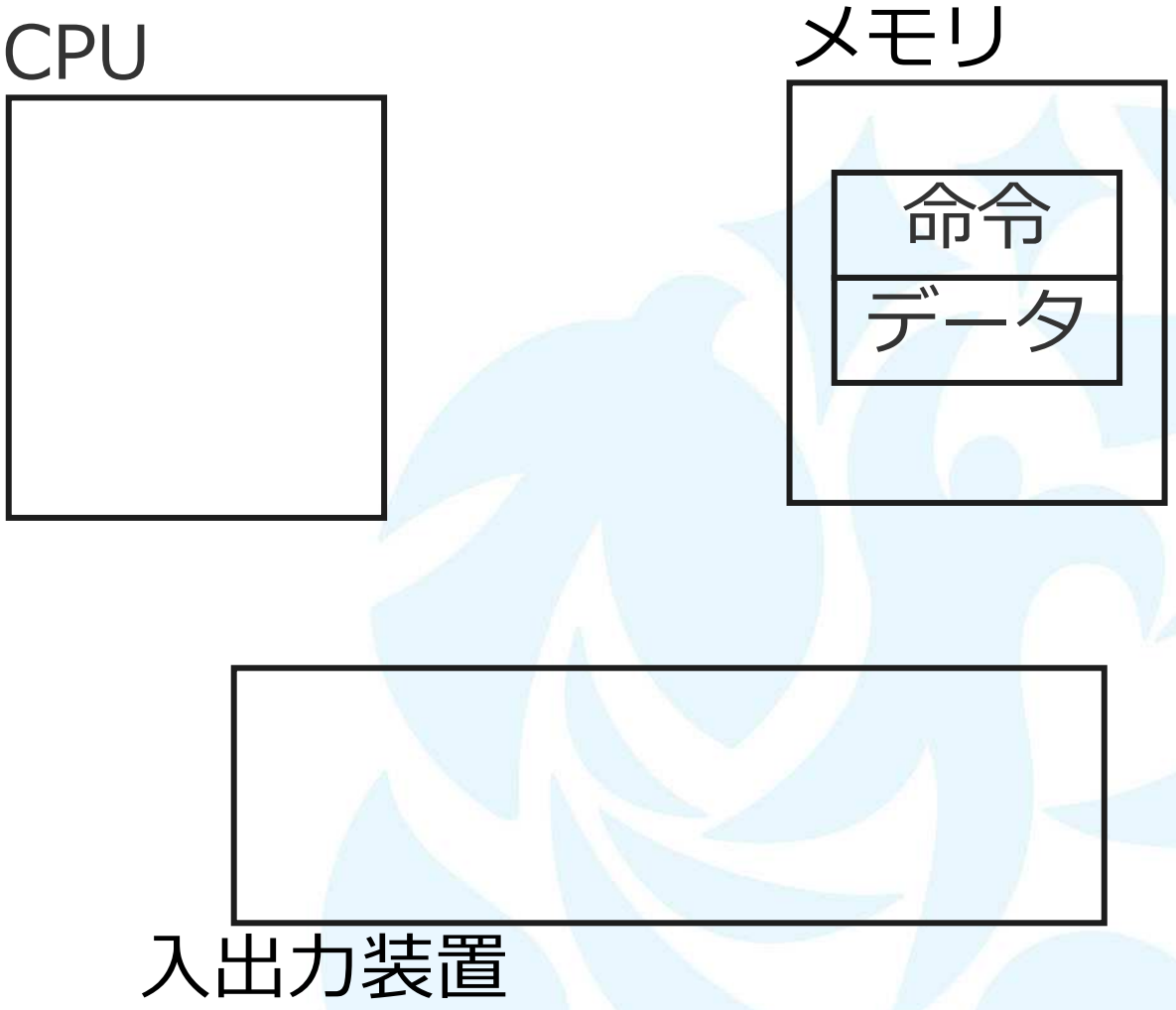
生命（いのち）の科学で未来をつなぐ

入出力アーキテクチャ

誰が入出力装置を制御するのか

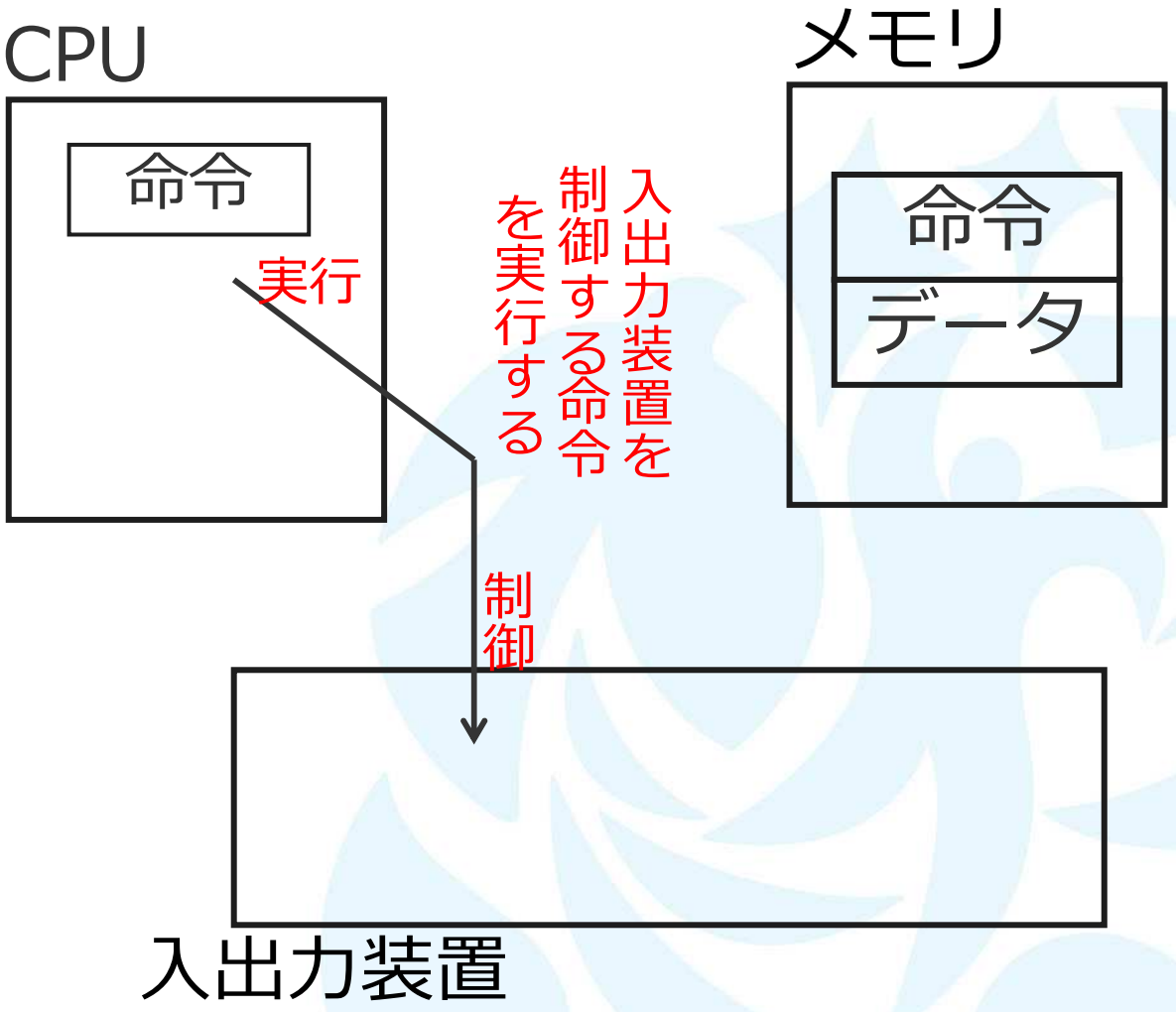
東邦大学

CPUが入出力装置を制御する



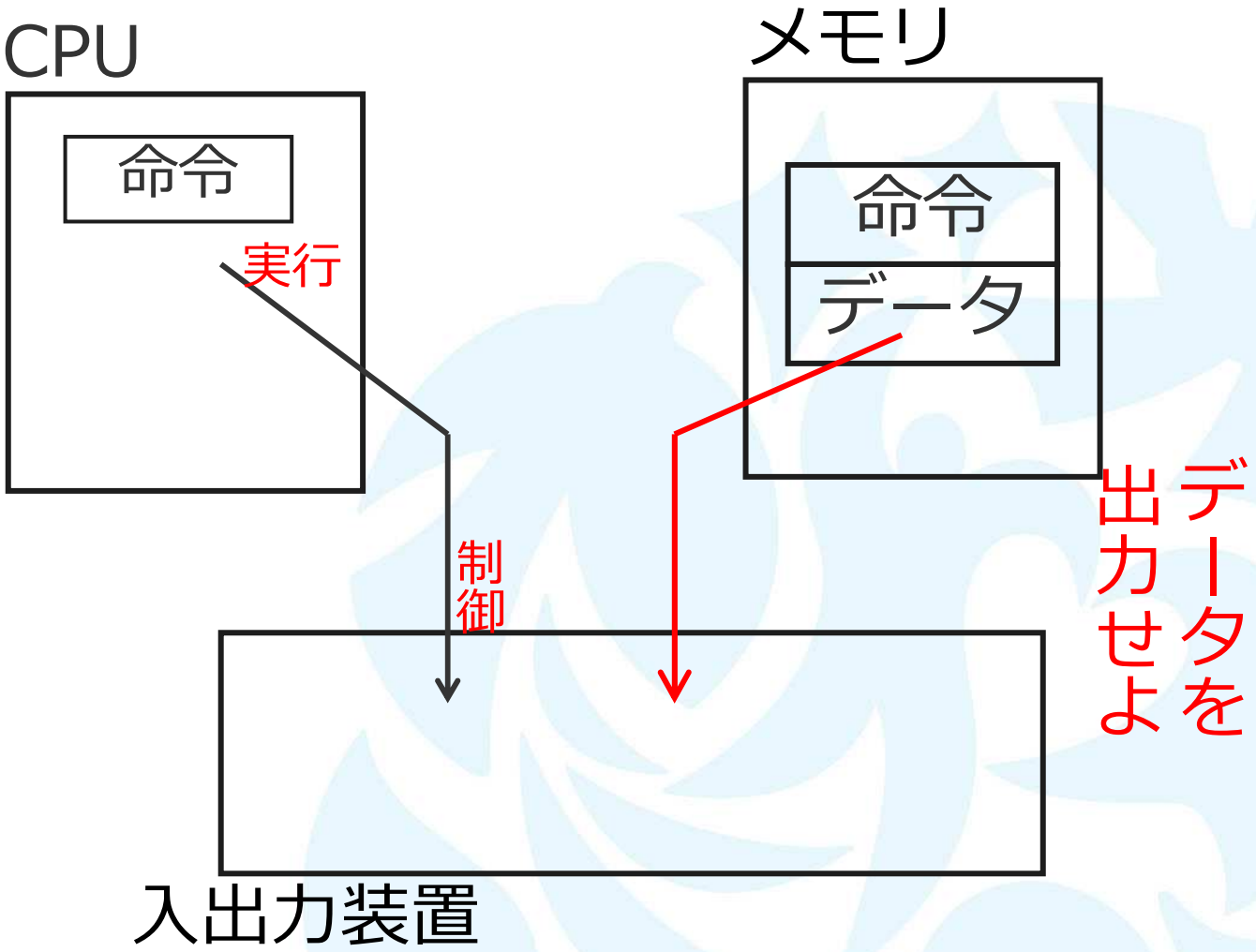
東邦大学

CPUが入出力装置を制御する

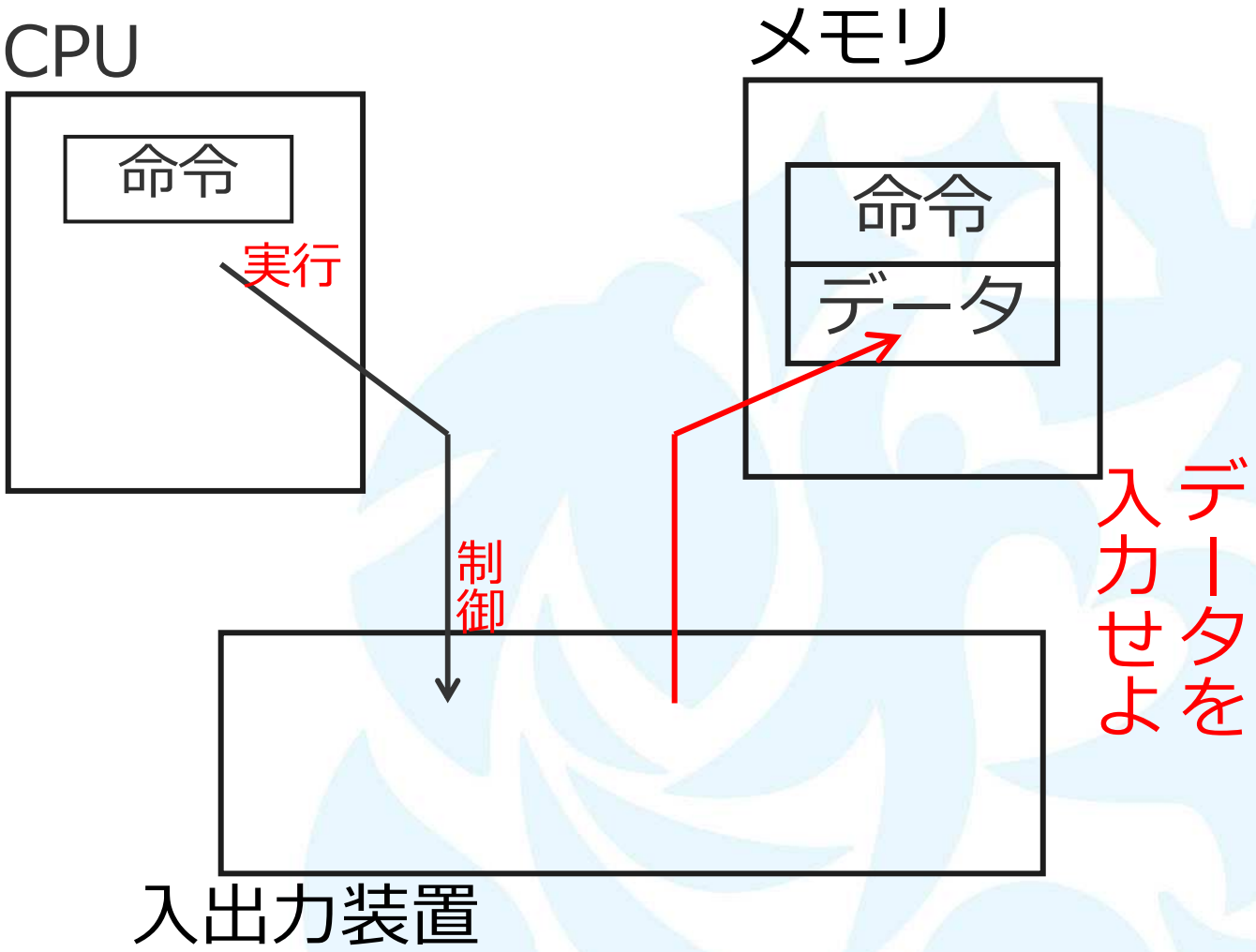


東邦大学

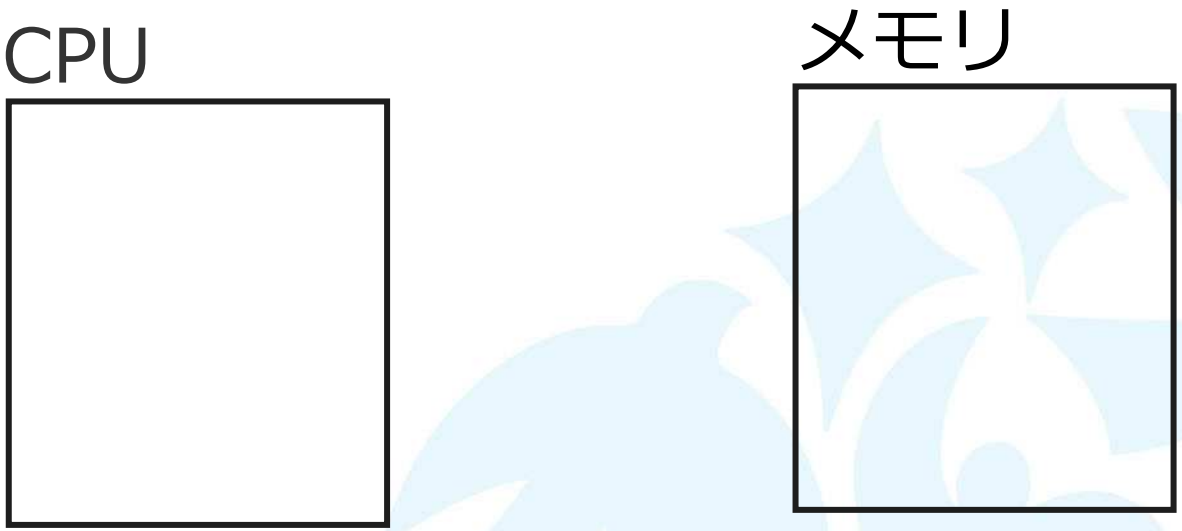
CPUが入出力装置を制御する



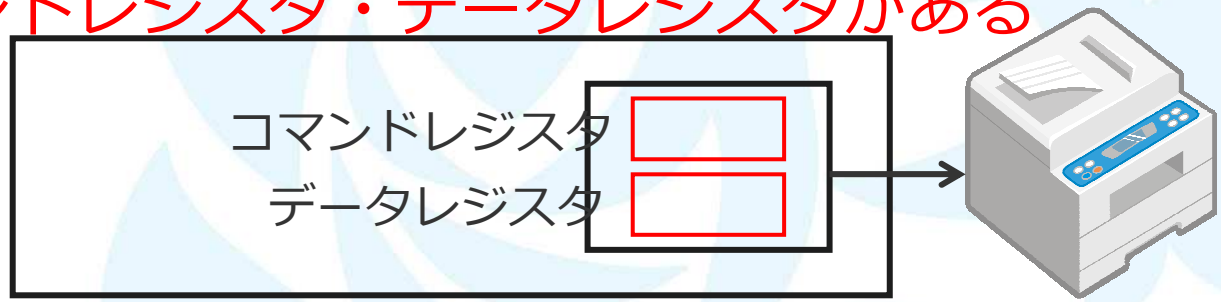
CPUが入出力装置を制御する



もう少し詳しく見ると



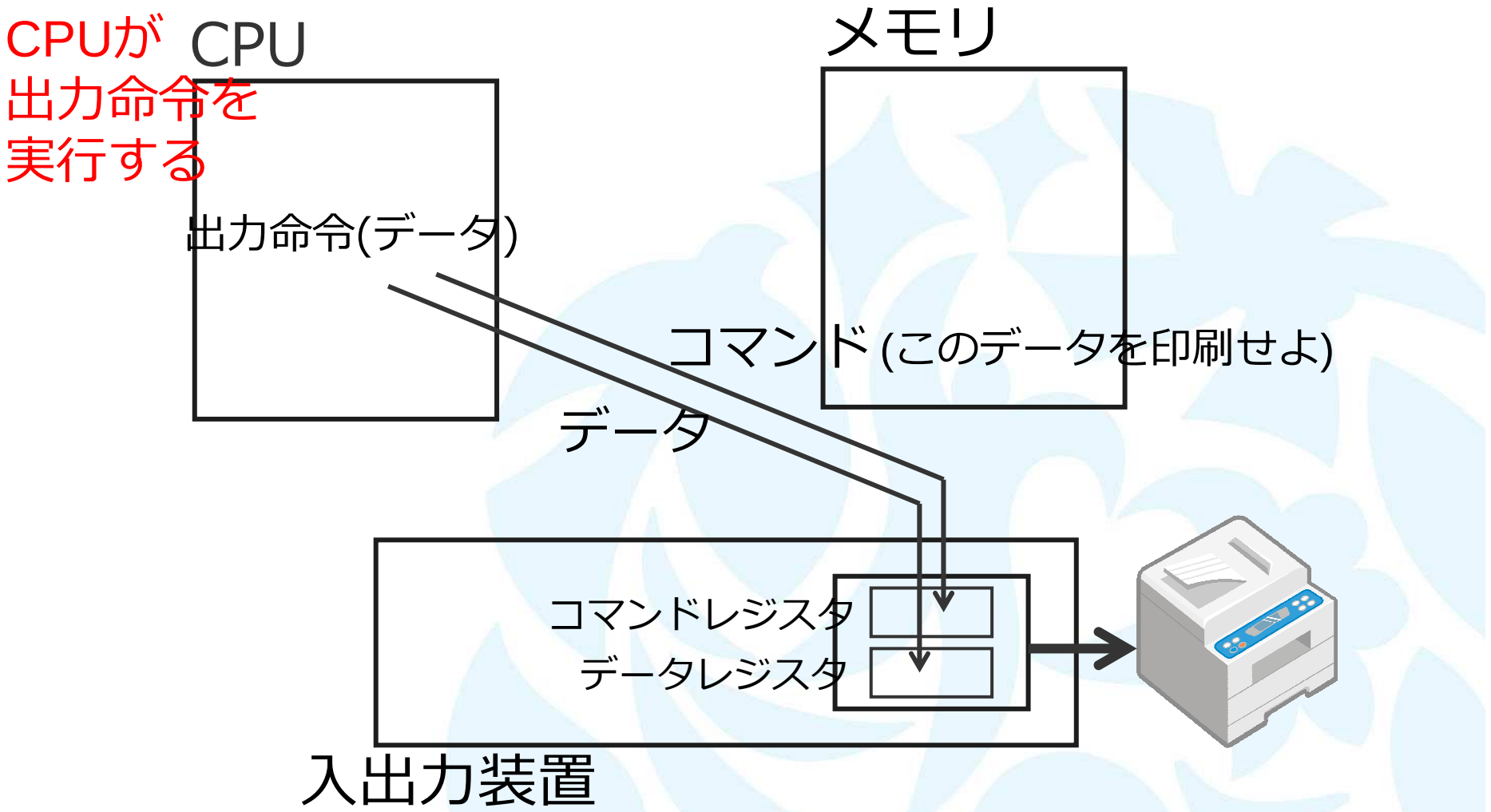
それぞれの入出力装置に接続されたアダプタがあり、
その中にコマンドレジスタ・データレジスタがある



入出力装置

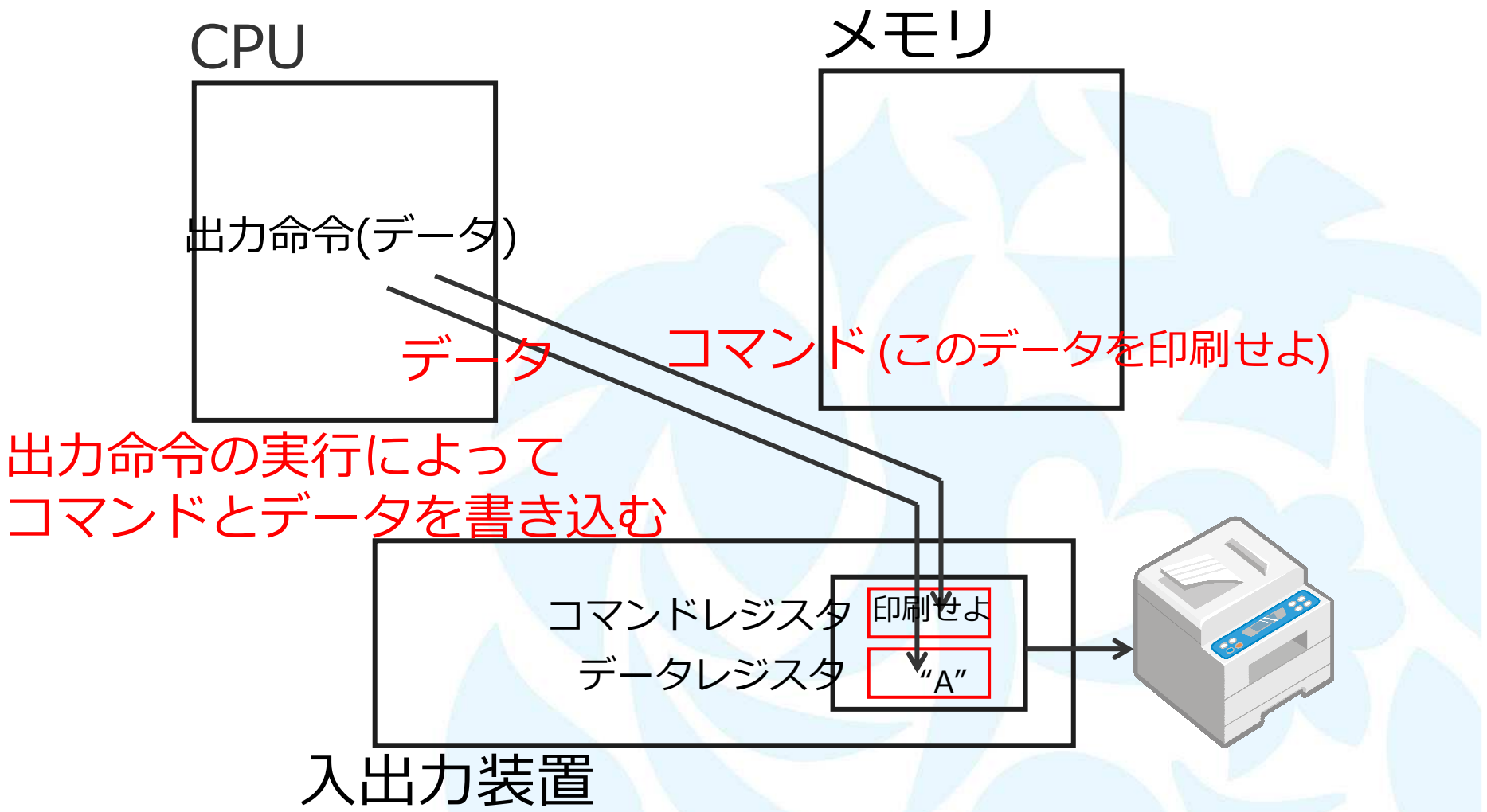
東邦大学

もう少し詳しく見ると

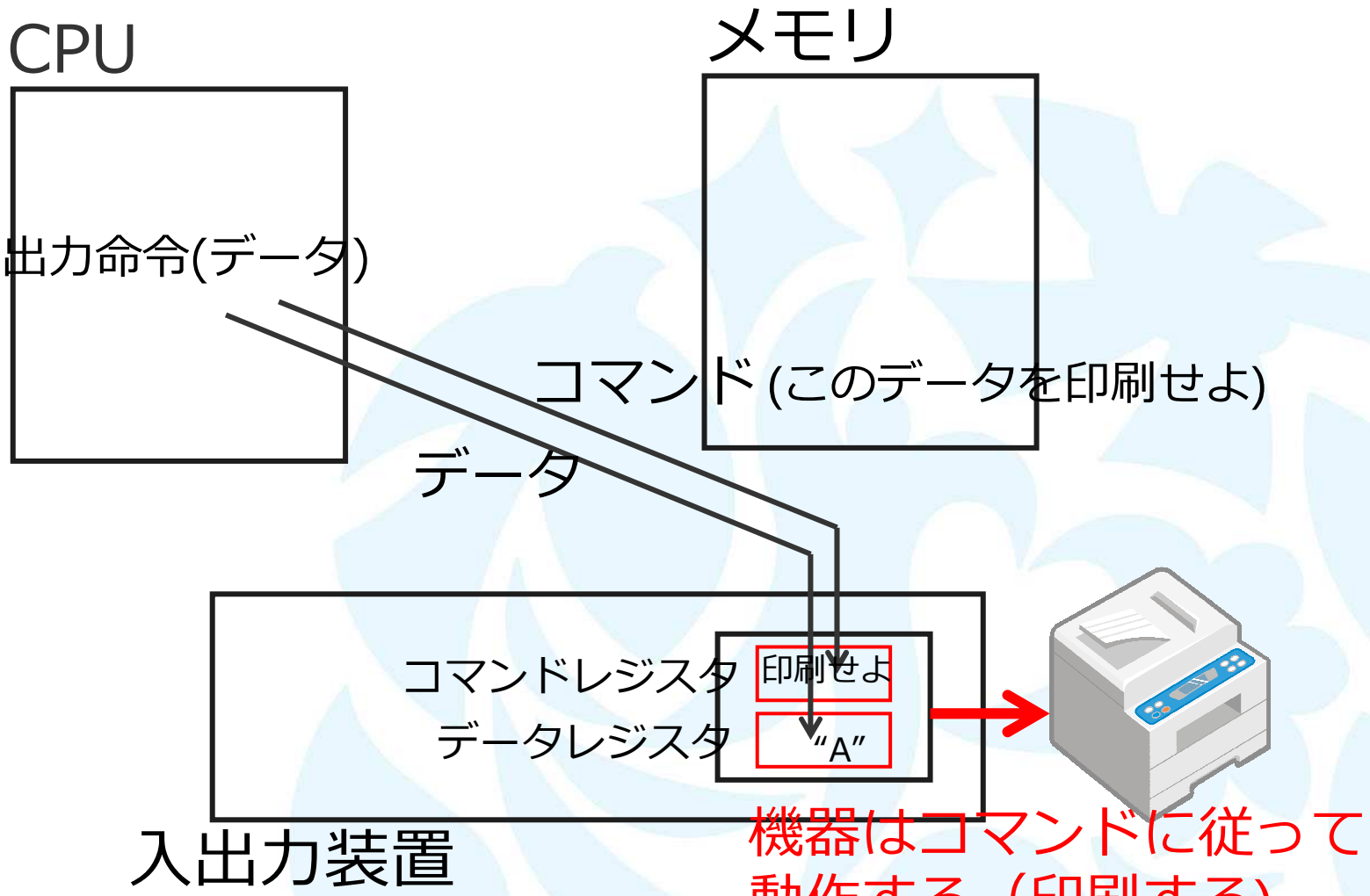


東邦大学

もう少し詳しく見ると



もう少し詳しく見ると



東邦大学

入出力アダプタのレジスタの指定

２つの指定方法（ハードの構造)がある

１）メモリマップト入出力

入出力アダプタのレジスタの指定

２つの指定方法（ハードの構造)がある

１）メモリマップト入出力

コマンド/データレジスタは、CPUから見るとメモリの一部（特定アドレスのメモリ）に見える
（メモリアドレス空間の一部にマップされる、と言う）

入出力アダプタのレジスタの指定

２つの指定方法（ハードの構造)がある

１）メモリマップト入出力

コマンド/データレジスタは、CPUから見るとメモリの一部（特定アドレスのメモリ）に見える
（メモリアドレス空間の一部にマップされる、と言う）

⇒　コマンド/データレジスタへの書込みは
　メモリの特定アドレスへのストア命令で行う

入出力アダプタのレジスタの指定

２つの指定方法（ハードの構造)がある

１）メモリマップト入出力

コマンド/データレジスタは、CPUから見るとメモリの一部（特定アドレスのメモリ）に見える
（メモリアドレス空間の一部にマップされる、と言う）

２）I/Oマップト入出力

コマンド/データレジスタは、CPUから見るとI/Oレジスタとして見える
（特別なI/Oレジスタ空間の一部にマップされる、と言う）

東邦大学

入出力アダプタのレジスタの指定

２つの指定方法（ハードの構造)がある

１）メモリマップト入出力

コマンド/データレジスタは、CPUから見るとメモリの一部（特定アドレスのメモリ）に見える
（メモリアドレス空間の一部にマップされる、と言う）

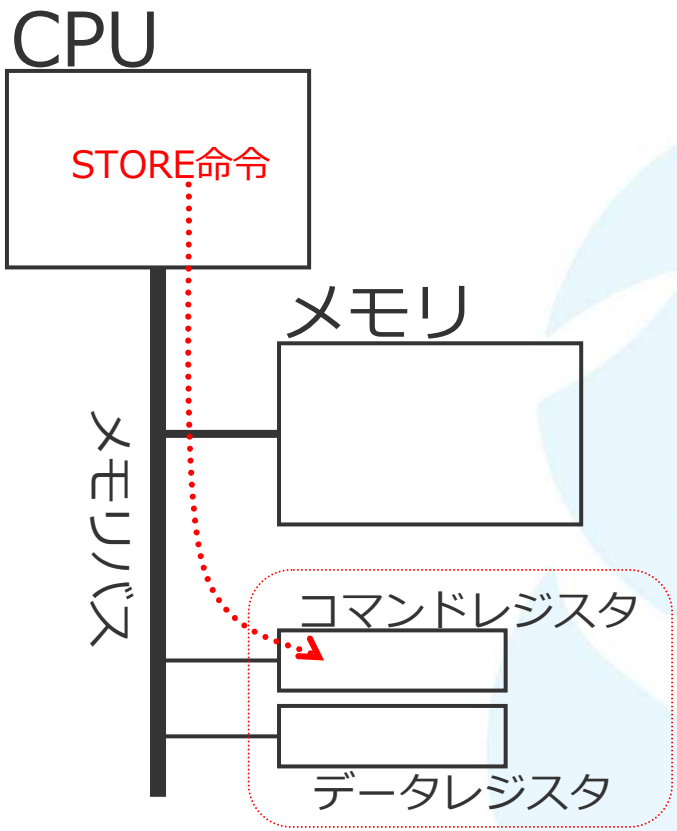
２）I/Oマップト入出力

コマンド/データレジスタは、CPUから見るとI/Oレジスタとして見える
（特別なI/Oレジスタ空間の一部にマップされる、と言う）

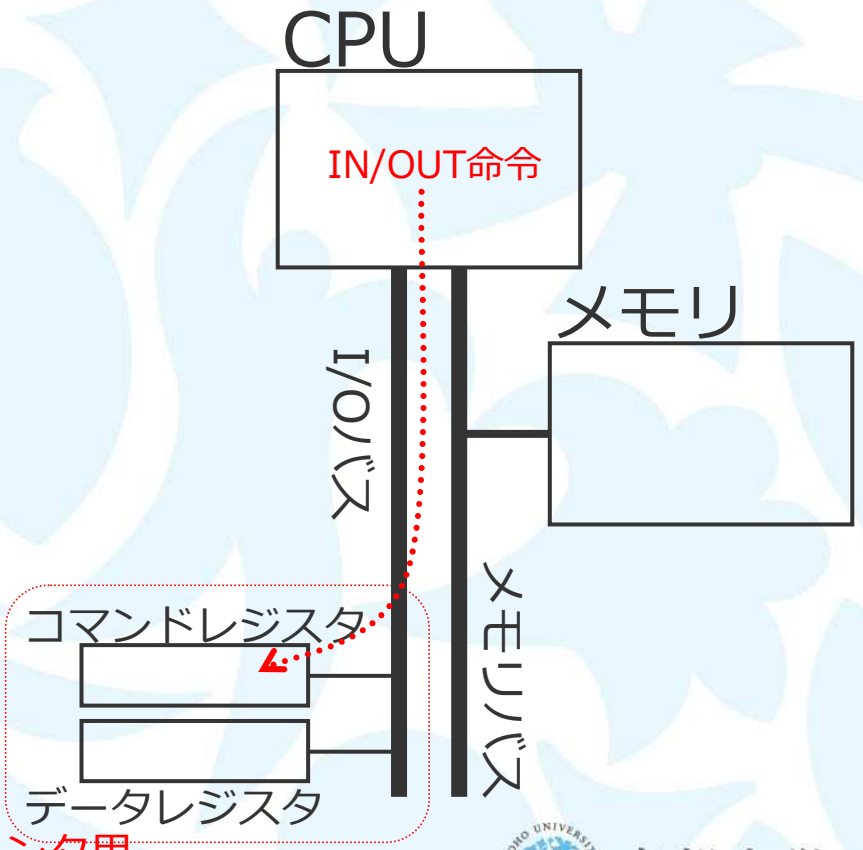
⇒　特別なI/O命令で書き込む

入出力アダプタのレジスタの指定

メモリマップト



I/Oマップト



データの転送方法が２つある

データ転送の方法

２つの転送方法（ハードの構造)がある

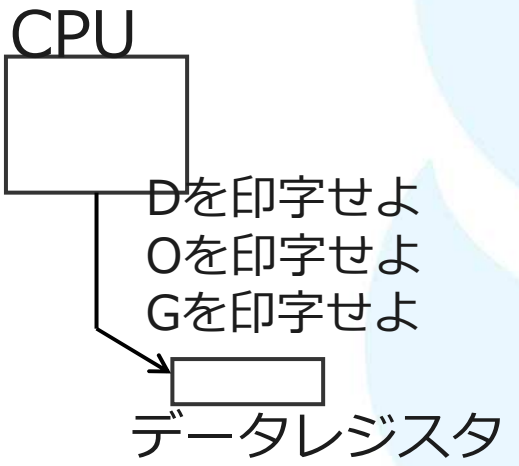
１）直接制御転送（プログラム制御転送）

データ転送の方法

２つの転送方法（ハードの構造)がある

１）直接制御転送（プログラム制御転送）

CPUが１バイトずつデータレジスタへ転送

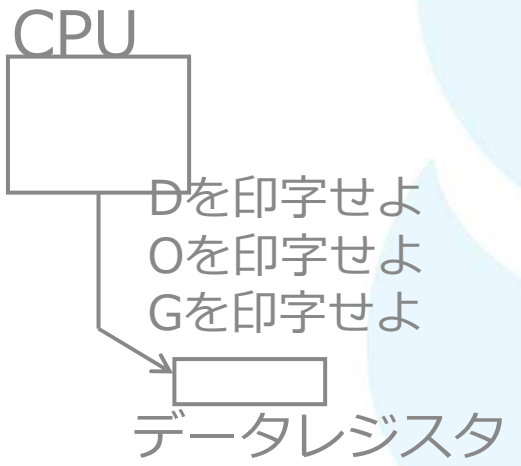


データ転送の方法

２つの転送方法（ハードの構造)がある

１）直接制御転送（プログラム制御転送）
　CPUが１バイトずつデータレジスタへ転送

２）間接制御転送（DMA転送）

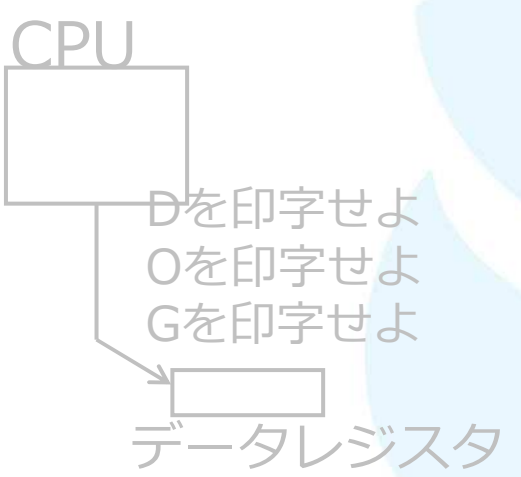


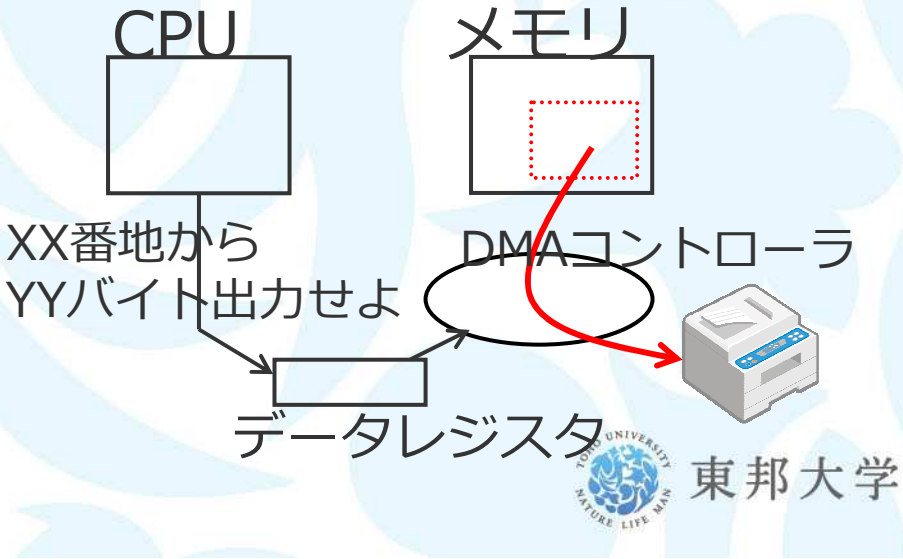
データ転送の方法

２つの転送方法（ハードの構造)がある

１）直接制御転送（プログラム制御転送）
CPUが１バイトずつデータレジスタへ転送

２）間接制御転送（DMA転送）
CPUは起動だけ指示。データ転送は入出力装置内のDMAコントローラが制御する





東邦大学

データ転送方法の比較

直接転送	間接転送

データ転送方法の比較

直接転送	間接転送
CPUが1バイト毎に関与する	CPUは転送開始時と終了時だけ関与する

データ転送方法の比較

直接転送	間接転送
CPUが1バイト毎に関与する	CPUは転送開始時と終了時だけ関与する
⇒CPU負荷が大きい ⇒低速の転送に向く	⇒CPU負荷が小さい ⇒高速の転送に向く

データ転送方法の比較

直接転送	間接転送
CPUが1バイト毎に関与する	CPUは転送開始時と終了時だけ関与する
⇒CPU負荷が大きい ⇒低速の転送に向く	⇒CPU負荷が小さい ⇒高速の転送に向く
余分なハード不要 ⇒安価に済む	DMAコントローラ要 その分ハード価格高

データ転送方法の比較

直接転送	間接転送
CPUが1バイト毎に関与する	CPUは転送開始時と終了時だけ関与する
⇒CPU負荷が大きい ⇒低速の転送に向く	⇒CPU負荷が小さい ⇒高速の転送に向く
余分なハード不要 ⇒安価に済む	DMAコントローラ要 その分ハード価格高
キーボード・プリンタ ディスプレイ・マウス	ハードディスク

間接転送制御の発展形

間接制御転送

直接制御転送　　DMA装置 ⇨

CPUが １バイトずつ 制御しながら 転送	CPUはDMA 装置へ開始番地 とバイト数を 指示して起動。 DMAが転送

間接転送制御の発展形

直接制御転送	間接制御転送 DMA装置 ⇨ チャネル装置	
CPUが １バイトずつ 制御しながら 転送	CPUはDMA 装置へ開始番地 とバイト数を 指示して起動。 DMAが転送	CPUは チャネルへ 一連の転送を 指示して起動。 チャネルが転送

間接転送制御の発展形

間接制御転送

直接制御転送　　DMA装置 ⇨ チャネル装置

CPUが １バイトずつ 制御しながら 転送	CPUはDMA 装置へ開始番地 とバイト数を 指示して起動。 DMAが転送	CPUは チャネルへ 一連の転送を 指示して起動。 チャネルが転送

複数・複雑な転送可能
（チャネルは小さなCPU）

東邦大学

間接転送制御の発展形

直接制御転送	間接制御転送 DMA装置 ⇨ チャネル装置	
CPUが １バイトずつ 制御しながら 転送	CPUはDMA 装置へ開始番地 とバイト数を 指示して起動。 DMAが転送	CPUは チャネルへ 一連の転送を 指示して起動。 チャネルが転送
		複数・複雑な転送可能 （チャネルは小さなCPU）
	比較的簡単なので PCで使われている	メインフレームで かつて使われた

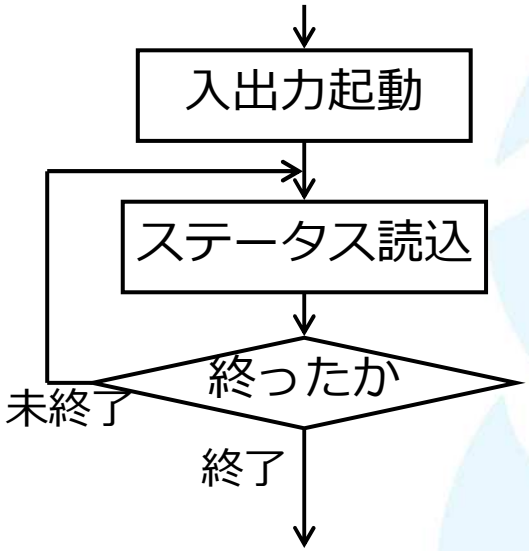
(脱線) 終了時のCPUの対応

入出力装置の動作終了をCPUが知るには？

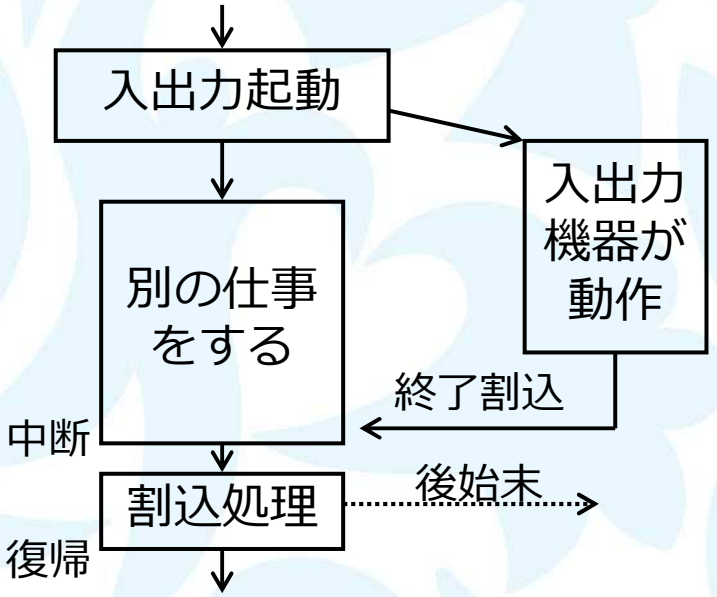
(脱線) 終了時のCPUの対応

入出力装置の動作終了をCPUが知るには？

終了をチェックし続ける



割込みによって知る



東邦大学

(脱線) 終了時のCPUの対応

終了をチェックし続ける

⇩

CPUは他のことができない

専用の機械ならこれもOK

プログラムは簡単

割込みによって知る

⇩

CPUは他の仕事ができる

PCや、多数の仕事をする機械ならこの方が普通

プログラムはやや複雑

入出力制御のまとめ

CPUからI/Oのコマンド/データレジスタに
情報を渡す（入力ならデータは読込む）

入出力制御のまとめ

CPUからI/Oのコマンド/データレジスタに
情報を渡す（入力ならデータは読込む）

渡す経路のスタイルとして、I/Oマップトと
メモリマップトがある

I/Oマップトとメモリマップトで、使う命令が違う

入出力制御のまとめ

CPUからI/Oのコマンド/データレジスタに
情報を渡す（入力ならデータは読込む）

渡す経路のスタイルとして、I/Oマップトと
メモリマップトがある

I/Oマップトとメモリマップトで、使う命令が違う

データの渡し方として、直接制御転送と
間接制御転送（DMA）がある

直接制御はハードが簡単だがCPU負荷大で低速用
間接制御は余分なハードが必要、CPU負荷小で高速用

東邦大学

入出力制御のまとめ

CPUからI/Oのコマンド/データレジスタに
情報を渡す（入力ならデータは読込む）

渡す経路のスタイルとして、I/Oマップトと
メモリマップトがある

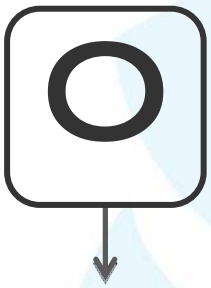
- I/Oマップトとメモリマップトで、使う命令が違う

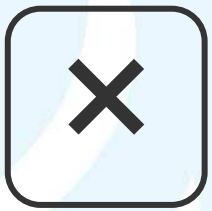
データの渡し方として、直接制御転送と
間接制御転送（DMA）がある

- 直接制御はハードが簡単だがCPU負荷大で低速用
- 間接制御は余分なハードが必要、CPU負荷小で高速用

理解できましたか？

入出力装置のつなぎ方が
分かりましたか？





東邦大学