

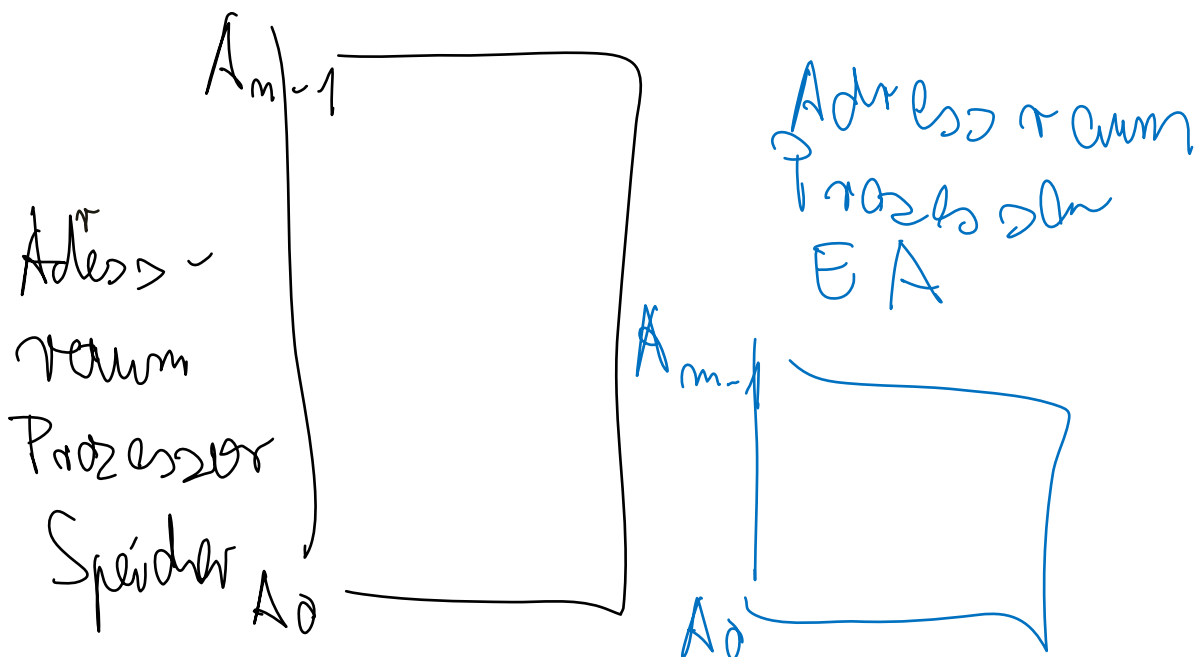
TI2 4.7.12

5. Ein-Ausgabe-Funktionseinheiten (EA, Input-Output, IO)

hier EA bis zur logisch elektrischen Schnittstelle (siehe F5_10)

➔ Aufgaben:

- physische Schnittstellen zur Ein- und Ausgabe von Eingangs- und Ausgangsoperanden (Befehle von außen sind i.a. beim Laden (z.B. von Festplatte noch Operanden)
- Unterscheidung von EA-Schnittstellen durch Adressierung (siehe auch F3_20, zumeist sind EA-Adressen und Speicheradressen getrennt)



Grundlage aller EA-Funktionen ist die digitale parallele EA

Datendarstellung:

F5_30:

Wert: Interpretation der binären Signale entsprechend Datentyp (Datendarstellung, siehe TI1)

verschiedene Werte folgen i.a. zeitlich hintereinander

F5_20 oben

Parallele digitale Ausgabe:

Register: speichert ein Datenwort (zum Zeitpunkt des Ausgabebefehls) bis zur nächsten Ausgabe
Datenwort wird über den Datenbus transportiert

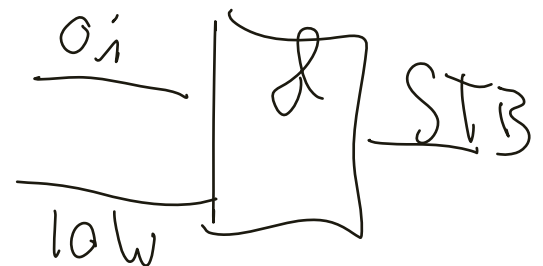
Dekoder: Erkennung, dass sich ein Ausgabebefehl auf die (hier dargestellte) aktuelle Schnittstelle bezieht, und dass es eine Ausgabe ist.

1-aus-n-Dekoder (siehe 4. Speicher, F4_70):

A_i (hier $i=m$) Adresssignale sind die Adresssignale des Prozessors (AB),

O_i^* sind Auswahlssignale für die EA-Schnittstellen

STB (einspeichern in das Register)
 $= O_i^* \wedge IOW$ (input output write)



O_i^* ist O_2^{i-j} (von Dekoder F4_70)

Ausgänge des Registers: Daten zur Umwelt (Peripherie)

Parallele digitale Eingabe (F5_20 unten)

Tristate-Treiber: Durchschalten der Daten der Umwelt (Peripherie) im IN-Befehl (Eingabebefehl) auf den Datenbus

Dekoder: Aufgabe sinngemäß wie bei der Ausgabe:

OE (Durchschaltsignal des Treibers)
=O_i (O_i siehe oben) \wedge IOR (input output read)

IOR, IOW sind Ausgangssignale des Prozessors im IN- bzw. OUT-Befehl

Serielle digitale Ein-Ausgabe:

Datendarstellung: F5_110

Ausgangspunkt: parallele digitale Datendarstellung:
ein bit eines Datenworts (DW entspricht Wert) ist einer Signalleitung zugeordnet (genau)

serielle digitale Datendarstellung:

Die einzelnen bit eines DW benutzen eine Signalleitung zeitlich nacheinander (F5_110 rechts), für n bit werden n Zeitpunkte notwendig, zeitlich aufeinander folgende DW in der parallelen Datendarstellung folgen auch in der seriellen zeitlich hintereinander, aber je n Zeitpunkte

Grundelement zur parallel-seriell-Wandlung oder umgekehrt:
Schieberegister (siehe TI1 bzw F5_130)

Parallelbetrieb: E-Daten ($D_{in,k}$) werden mit Takt in des Register gespeichert, Ausgabedaten liegen immer bis zum nächsten Speichern an

Seriellbetrieb: E-Daten ($D_{in,k}$) sind Ausgabedaten $D_{out,k+1}$ (bzw. seriell in), werden mit jedem Takt gespeichert und „wandern“ damit mit jedem Takt ein Register weiter

Parallel-Seriellwandlung: Parallel Speichern, seriell heraus schieben (über seriell out)

Seriell-Parallelwandlung (Seriell Einschreiben (über seriell in) und Parallel Lesen

Vollständige Ein-Ausgabe (seriell) -> F5_140)

Ablauf:

- Datenwort (DW) über parallel Ausausgeben und parallel in das Schieberegister (SR_{links}) übernehmen)
- Seriell heraus schieben mit Takt vom Taktgenerator ($links$)
- Seriell einschreiben in SR ($rechts$) mit Takt vom Taktgenerator ($rechts$)
- DW über parallel Ein parallel vom SR lesen

Restliche Blöcke: logisch zeitliche Synchronisation

Takt ($links, Aus$) = Takt ($rechts, Ein$) (erste Näherung)

damit herausgeschobene bit zum richtigen Zeitpunkt eingeschoben werden.

Analoge Eingabe (Analoge Ausgabe recht selten, deshalb hier nicht betrachtet)

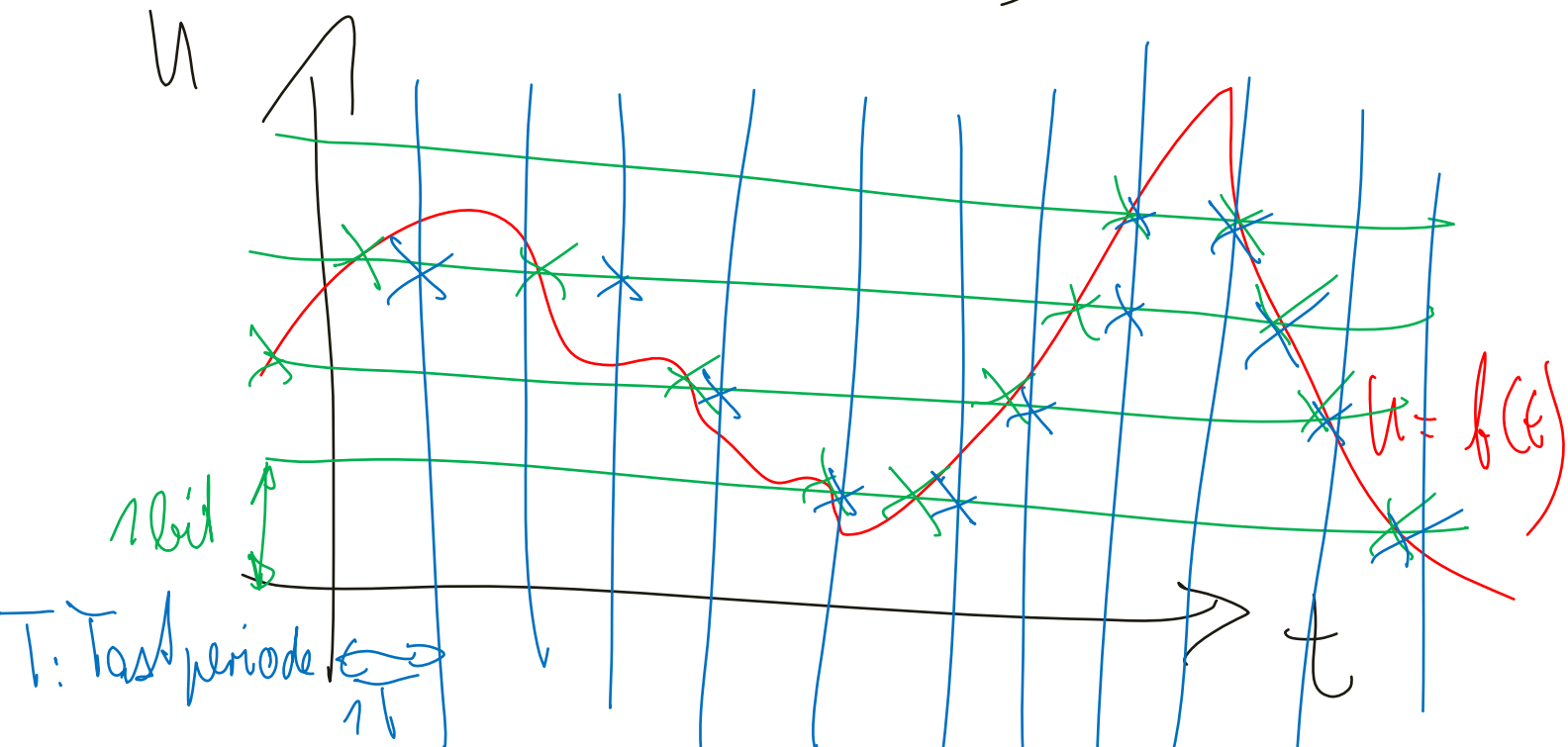
Datendarstellung:

Wert=f(Zeit)

hier meist:

- Wert Spannung (kontinuierlich)
- Zeit (kontinuierlich)

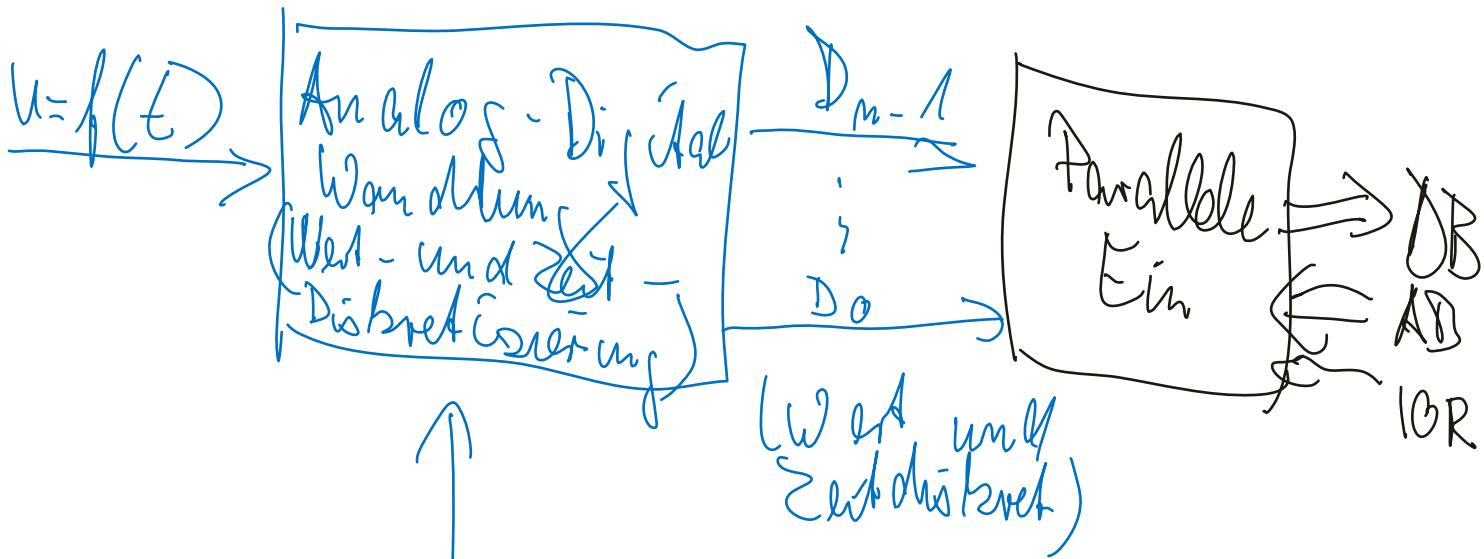
(u)
(t)



Wertquantifizierung, hier Abbildung der analogen Spannung auf einen digitale (Festkomma-)Zahl \rightarrow x Werte

Zeitquantifizierung, hier Abbildung der kontinuierlichen Zeit auf diskrete Zeitpunkte \rightarrow x Werte wert diskret

und zeitdiskret



verschiedene Prinzipien
und Verfahren