



(10) **DE 102 02 090 B4** 2010.08.12

(12) **Patentschrift**

(21) Aktenzeichen: **102 02 090.6**
(22) Anmeldetag: **21.01.2002**
(43) Offenlegungstag: **31.07.2003**
(45) Veröffentlichungstag
der Patenterteilung: **12.08.2010**

(51) Int Cl.⁸: **H03M 13/00** (2006.01)
H04L 27/34 (2006.01)

Innerhalb von drei Monaten nach Veröffentlichung der Patenterteilung kann nach § 59 Patentgesetz gegen das Patent Einspruch erhoben werden. Der Einspruch ist schriftlich zu erklären und zu begründen. Innerhalb der Einspruchsfrist ist eine Einspruchsgebühr in Höhe von 200 Euro zu entrichten (§ 6 Patentkostengesetz in Verbindung mit der Anlage zu § 2 Abs. 1 Patentkostengesetz).

(73) Patentinhaber:
Infineon Technologies AG, 81669 München, DE

(74) Vertreter:
**Graf Lambsdorff, M., Dipl.-Phys.Dr.rer.nat.,
Pat.-Anw., 81673 München**

(72) Erfinder:
**Marsili, Stefano, Fürnitz, AT; Bacher, Martin,
Villach, AT**

(56) Für die Beurteilung der Patentfähigkeit in Betracht
gezogene Druckschriften:

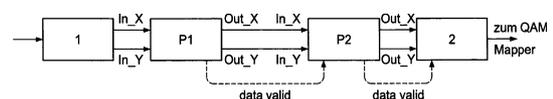
DE 195 24 556 A1
DE 100 48 872 A1
US 2001/00 34 868 A1
EP 03 01 161 B1

**SWEENEY P.: Codierung zur Fehlererkennung
und Fehlerkorrektur. München, Hanser, 1992.
ISBN: 3-446-16439-1, S. 144-146**

(54) Bezeichnung: **Elektronische Sender-Empfänger-Vorrichtung**

(57) Hauptanspruch: Punktierereinrichtung zur Verwendung in einer elektronischen Sendervorrichtung, wobei die Punktierereinrichtung

- einen ersten und einen zweiten Datenausgang aufweist,
- so ausgelegt ist, daß sie ihren Ausgabedatenstrom im Wesentlichen gleichmäßig parallel auf ihre beiden genannten Datenausgänge verteilt,
- leere Plätze in ihrem Ausgabedatenstrom vorsieht, so dass die Anzahl der Bits des Eingangsdatenstroms der Anzahl der Bits des Ausgangsdatenstroms einschließlich der leeren Plätze entspricht, und
- zusätzlich zu ihrem parallelen Ausgabedatenstrom ein Signal (data_valid) ausgibt, welches leere Plätze im parallelen Ausgabedatenstrom der Punktierereinrichtung anzeigt.



Beschreibung

[0001] Die Erfindung betrifft eine Punktierereinrichtung zur Verwendung in einer elektronischen Sendervorrichtung nach dem Anspruch 1, einen Verschachteler zur Verwendung in einer elektronischen Sendervorrichtung nach dem Anspruch 10, einen Entschachteler zur Verwendung in einer elektronischen Empfängervorrichtung nach dem Anspruch 21 und eine Depunktierereinrichtung zur Verwendung in einer elektronischen Empfängervorrichtung nach dem Anspruch 33.

[0002] Allgemein gesprochen betrifft die Erfindung solche elektronischen Nachrichtenübertragungssysteme, bei denen senderseitig eine Datenpunktierung und/oder Datenverschachtelung und empfängerseitig eine Datenentschachtelung und/oder Datendepunktierung erfolgt oder zumindest teilweise erfolgt.

[0003] Derartiges geschieht beispielsweise sowohl im Rahmen des HIPERLAN/2(High Performance Radio Local Area Network Type 2)-Standards („ETSI TS 101 761-1 Broadband Radio Access Networks; Hiperlan Type 2; Physical Layer“) als auch im Rahmen des Standards „IEEE 802.11a – Part 11: Wireless LAN (WLAN) Medium Access Control and Physical Layer specifications: High-speed Physical Layer in the 5 GHz Band“. Neben den genannten Originalstandards können Angaben zum HIPERLAN/2-Standard im Internet unter www.hiperlan2.com bezogen werden. Ein Überblick über den HIPERLAN/2-Standard findet sich ferner in dem Artikel „HIPERLAN type 2 for broadband wireless communication“ von J. Khun-Jush et al. in Ericsson Review No. 2, 2000, Seiten 108 bis 119.

[0004] In beiden genannten Standards ist ein ähnliches Übertragungsfehlerkorrekturschema definiert. Es beinhaltet senderseitig (siehe [Fig. 2](#))

- 1) einen Faltungskodierer **1** mit der Koderate 1/2,
- 2) einen ersten Punktierer P1 zur Redundanzreduktion von 12 Bit (nur bei ETSI HIPERLAN/2),
- 3) einen zweiten Punktierer P2 zur Koderatenauswahl (Koderaten: 1/2, 9/16, 2/3, 3/4) und
- 4) einen Verschachteler **2** zur Vermeidung von Burstfehlern

sowie empfängerseitig (siehe [Fig. 3](#)) einen entsprechenden Entschachteler **3** und entsprechende Depunktierer P2', P1'.

[0005] Das Punktieren mit dem ersten Punktierer P1 ist nur im oben genannten ETSI-Standard definiert. Die P2-Rate von 2/3 kommt nur im oben genannten IEEE-Standard vor, die P2-Rate von 9/16 nur in dem ETSI-Standard. Die folgende Beschreibung bezieht sich im Wesentlichen auf den genannten ETSI-Standard und auf den genannten IEEE-Standard. Es sei jedoch ganz generell klargestellt, dass sich die vorliegende Erfindung nicht auf die Anwendung bzgl. der beiden genannten Standards beschränkt, sondern vielmehr allgemein auf alle Sender-Empfänger-Vorrichtungen, bei denen das zu übertragende Signal nach seiner Kodierung punktiert und/oder verschachtelt bzw. vor seiner Dekodierung entschachtelt und/oder depunktiert wird, anwendbar ist.

[0006] Der Kodierer **1** erzeugt für jedes Eingabebit zwei parallele Ausgabebits. Der erste Punktierer P1 entfernt einige dieser Ausgabebits des Kodierers **1** nach einem Schema gemäß der nachfolgenden Tabelle 1. Bei Bedarf wird dieses Punktierungsschema auf eine Gruppe von 156 kodierten Bits angewendet.

Tabelle 1

Bit-Nummerierung	Punktierungsmuster	Übertragene Sequenz (nach Parallel-Seriell-Wandlung)
0–155	X: 1111110111111 Y: 11111111111110	$X_0 Y_0 X_1 Y_1 X_2 Y_2 X_3 Y_3 X_4 Y_4 X_5 Y_5 X_7 Y_6$ $X_8 Y_7 X_9 Y_8 X_{10} Y_9 X_{11} Y_{10} X_{12} Y_{11}$
> 156 oder bei IEEE	X: 1 Y: 1	$X_0 Y_0$

[0007] Zwischen der Eingabe und der Ausgabe des ersten Punktierers P1 besteht eine Geschwindigkeitsdifferenz. Für eine Gruppe von 13 Bits z. B. erzeugt der Kodierer **1** 2×13 Bits, während die Ausgabe des ersten Punktierers P1 24 Bits lang ist. Deshalb werden, um die unterschiedlichen Geschwindigkeiten zwischen dem Kodiererausgang und dem Punktiererausgang einander anzupassen, FIFO-Speicherelemente in bestimmten Abschnitten vorgesehen, z. B. vor oder nach dem ersten Punktierer P1 oder sogar vor dem Faltungskodierer **1**.

[0008] Eine Besonderheit der oben erwähnten Standards besteht darin, daß jeder möglichen Datenrate eine bestimmte Modulationsform sowie eine bestimmte Koderate (für die Kanalkodierung) zugeordnet ist. Die Zuordnung ist so gewählt, daß die Anzahl der kodierten Bits pro OFDM(Orthogonal Frequency Division Multiplexing)-Symbol, bezeichnet mit NCBPS, stets ein Vielfaches der Anzahl der Subkanäle (48 Stück) ist. Tabelle 3 gibt die Spezifikation der Übertragungsmodi wieder. Es wird deutlich, daß eine BPSK-(Binary Phase Shift Keying-) und eine QPSK-(Quarternary Phase Shift Keying-)Modulation sowie zwei Amplitudenmodulationen (16 QAM: 16-ary-Quadrature Amplitude Modulation; 64 QAM: 64-ary-Quadrature Amplitude Modulation) eingesetzt werden. Als Koderate r wird das Verhältnis der Anzahl der unkodierten Bits zu der Anzahl der kodierten Bits bezeichnet.

[0009] Im schnellsten Modus (54 Mbps, 64 QAM), wenn sowohl die erste Punktierung P1 als auch die zweite Punktierung P2 (Koderate 3/4) angewendet werden, müssen 222 Eingabebits für ein OFDM-Symbol kodiert werden (siehe Tabelle 3). Daher würde man 2×222 Bits am Ausgang des Kodierers 1 und 1×432 Bits am Ausgang des ersten Punktierers P1 haben.

[0010] Gemäß der oben erwähnten Standard-Definitionen wird der Ausgangsdatenstrom des ersten Punktierers P1 wieder seriellparallel (S/P) gewandelt, ehe er dem zweiten Punktierer P2 zugeführt wird.

[0011] Der zweite Punktierer P2 entfernt wieder einige der Eingangsbits, und zwar gemäß dem in der nachfolgenden Tabelle 2 dargestellten Schema.

Tabelle 2

Koderaten r	Punktierungsmuster	Übertragene Sequenz (nach Parallel-Seriell-Wandlung)
1/2	X: 1 Y: 1	$X_0 Y_0$
9/16 (nur bei ETSI)	X: 111111110 Y: 111101111	$X_0 Y_0 X_1 Y_1 X_2 Y_2 X_3 Y_3 X_4 X_5 Y_5 X_6 Y_6 X_7 Y_7 Y_8$
2/3 (nur bei IEEE)	X: 11 Y: 10	$X_0 Y_0 X_1$
3/4	X: 110 Y: 101	$X_0 Y_0 X_1 Y_2$

[0012] Für die zweite Punktierung kann man die gleichen Betrachtungen bzgl. des Geschwindigkeitsunterschieds anstellen, wie sie oben für das erste Punktieren gemacht wurden. Man benötigt daher wiederum ein FIFO-Speicherelement.

[0013] Der serielle Ausgangsdatenstrom des zweiten Punktierers P2 wird schließlich an den Verschachteler 2 geschickt, der die Bits nach einem Doppelschema umordnet. Das Verschachteln ist nur innerhalb eines OFDM-Symbols entsprechend einem Doppelschema definiert.

[0014] Im Folgenden sei k der Index für das kodierte Bit vor der ersten Permutation, i der Index nach der ersten und vor der zweiten Permutation und j der Index nach der zweiten Permutation unmittelbar vor der Modulationszuordnung (dem "Mapping").

[0015] Die erste Permutation wird durch folgende Gleichung 1 definiert:

$$i = \frac{NCBPS}{16} \cdot (k \bmod 16) + \text{floor}\left(\frac{k}{16}\right) \tag{Gl. 1}$$

mit $k = 0, 1, \dots, NCBPS - 1$

[0016] Die Funktion $\text{floor}(\cdot)$ bezeichnet die größte natürliche Zahl, die nicht den Parameter übersteigt, mod ist der natürliche Modulo-Operator, und NCBPS bezeichnet die Anzahl der kodierten Bits pro OFDM-Symbol (NCBPS = Number of Coded Bits Per OFDM Symbol).

[0017] Die zweite Permutation wird durch folgende Gleichung 2 definiert:

$$j = s \times \text{floor}\left(\frac{L}{s}\right) + (i + \text{NCBPS} - \text{floor}\left(16 \times \frac{i}{\text{NCBPS}}\right)) \bmod s \tag{Gl. 2}$$

mit $i = 0, 1, \dots, \text{NCBPS} - 1$

[0018] Der Wert von s wird durch die Anzahl der kodierten Bits pro Subträger, NBPS ($=$ Number of coded Bits Per Sub-Carrier), gemäß folgender Gleichung 3

$$s = \max\left(\frac{\text{NBPS}}{2}, 1\right) \tag{Gl. 3}$$

bestimmt.

[0019] Tabelle 3 faßt die Werte von NBPS und NCBPS für die verschiedenen Übertragungsmodi zusammen mit NUBPS ($=$ Number of Uncoded Bits Per OFDM Symbol, Anzahl der unkodierten Bits pro OFDM-Symbol) zusammen, und zwar sowohl für den Fall, daß der in [Fig. 1](#) dargestellte erste Punktierer P1 zum Einsatz kommt als auch für den Fall, daß der erste Punktierer P1 nicht angewendet wird.

Tabelle 3: Hauptparameter für verschiedene Übertragungsmodi

Übertragungsmodus (Modulationsart und Koderate)	N_{BPS}	N_{UBPS} P1 nicht angewendet	N_{UBPS} P1 angewendet	N_{CBPS}	Verschachtelter Operationsmodus
BPSK 1/2	1	24	26	48	0
BPSK 3/4	1	36	39	48	0
QPSK 1/2	2	48	52	96	1
QPSK 3/4	2	72	78	96	1
16 QAM 1/2	4	96	102	192	2
16 QAM 9/16	4	108	112	192	2
16 QAM 3/4	4	144	150	192	2
64 QAM 2/3	6	192	198	288	3
64 QAM 3/4	6	216	222	288	3

[0020] Empfängerseitig ([Fig. 3](#)) müssen die inversen Funktionen ausgeführt werden. In diesem Fall ist jedem Bit ein Zuverlässigkeitsmaß für den Viterbi-Dekodierer 4 zugeordnet, das durch $N - 1$ Bits repräsentiert wird. Man kann sagen, daß jedes Bit durch N Softbits dargestellt wird, wobei ein Softbit ein Wort ist (siehe [Fig. 11](#)).

[0021] Bei der Depunktierungsfunktion werden in den empfangenen Datenstrom dort Soft-Nullen eingefügt, wo senderseitig Bits entfernt worden waren. Dies wird nach dem Entschachtelerausgang gemäß den Tabellen 4 und 5 entsprechend durch den ersten Depunktierer P2', der in seiner Funktionsweise der Umkehrung des zweiten Punktierers P2 entspricht, und durch den zweiten Depunktierer P1', der in seiner Funktionsweise der Umkehrung des ersten Punktierers P1 entspricht, durchgeführt.

Tabelle 4: Depunktierungsschema des ersten Depunktierers P2'

Koderaten r	Empfangene Sequenz	Ausgangssequenz
1/2	$X_0 Y_0$	X: X_0 Y: Y_0
9/16	$X_0 Y_0 X_1 Y_1 X_2 Y_2 X_3 Y_3 X_4 X_5 Y_5 X_6 Y_6 X_7 Y_7 Y_8$	X: $X_0 X_1 X_2 X_3 X_4 X_5 X_6 X_7 0$ Y: $Y_0 Y_1 Y_2 Y_3 0 Y_5 Y_6 Y_7 Y_8$
2/3	$X_0 Y_0 X_1$	X: $X_0 X_1$ Y: $Y_0 0$
3/4	$X_0 Y_0 X_1 Y_2$	X: $X_0 X_1 0$ Y: $Y_0 0 Y_2$

Tabelle 5: Depunktierungsschema des zweiten Depunktierers P1'

Bit-Nummerierung	Empfangene Sequenz	Ausgangssequenz
0-155	$X_0 Y_0 X_1 Y_1 X_2 Y_2 X_3 Y_3 X_4 Y_4 X_5 Y_5 X_7 Y_6$ $X_8 Y_7 X_9 Y_8 X_{10} Y_9 X_{11} Y_{10} X_{12} Y_{11}$	X: $X_0 X_1 X_2 X_3 X_4 X_5 0 X_7 X_8 X_9 X_{10} X_{11} X_{12}$ Y: $Y_0 Y_1 Y_2 Y_3 Y_4 Y_5 Y_6 Y_7 Y_8 Y_9 Y_{10} Y_{11} 0$
> 156	$X_0 Y_0$	X: X_0 Y: Y_0

[0022] Wie auf der Senderseite, so ergeben sich auch hier die gleichen Probleme hinsichtlich Latenz und Geschwindigkeitsanpassung.

[0023] Konzentrieren wir uns nun auf die erste Permutation (Gleichung 1). In Tabelle 6 sind die 288 Eingabe-/Ausgabebits für den 64 QAM Modus dargestellt.

Tabelle 6

Modus #3											
k	i	k	i	k	i	k	i	k	i	k	i
0	0	48	3	96	6	144	9	192	12	240	15
1	18	49	21	97	24	145	27	193	30	241	33
2	36	50	39	98	42	146	45	194	48	242	51
3	54	51	57	99	60	147	63	195	66	243	69
4	72	52	75	100	78	148	81	196	84	244	87
5	90	53	93	101	96	149	99	197	102	245	105
6	108	54	111	102	114	150	117	198	120	246	123
7	126	55	129	103	132	151	135	199	138	247	141
8	144	56	147	104	150	152	153	200	156	248	159
9	162	57	165	105	168	153	171	201	174	249	177
10	180	58	183	106	186	154	189	202	192	250	195
11	198	59	201	107	204	155	207	203	210	251	213
12	216	60	219	108	222	156	225	204	228	252	231
13	234	61	237	109	240	157	243	205	246	253	249
14	252	62	255	110	258	158	261	206	264	254	267
15	270	63	273	111	276	159	279	207	282	255	285
16	1	64	4	112	7	160	10	208	13	256	16
17	19	65	22	113	25	161	28	209	31	257	34
18	37	66	40	114	43	162	46	210	49	258	52
19	55	67	58	115	61	163	64	211	67	259	70
20	73	68	76	116	79	164	82	212	85	260	88
21	91	69	94	117	97	165	100	213	103	261	106
22	109	70	112	118	115	166	118	214	121	262	124
23	127	71	130	119	133	167	136	215	139	263	142
24	145	72	148	120	151	168	154	216	157	264	160
25	163	73	166	121	169	169	172	217	175	265	178
26	181	74	184	122	187	170	190	218	193	266	196
27	199	75	202	123	205	171	208	219	211	267	214
28	217	76	220	124	223	172	226	220	229	268	232

29	235	77	238	125	241	173	244	221	247	269	250
30	253	78	256	126	259	174	262	222	265	270	268
31	271	79	274	127	277	175	280	223	283	271	286
32	2	80	5	128	8	176	11	224	14	272	17
33	20	81	23	129	26	177	29	225	32	273	35
34	38	82	41	130	44	178	47	226	50	274	53
35	56	83	59	131	62	179	65	227	68	275	71
36	74	84	77	132	80	180	83	228	86	276	89
37	92	85	95	133	98	181	101	229	104	277	107
38	110	86	113	134	116	182	119	230	122	278	125
39	128	87	131	135	134	183	137	231	140	279	143
40	146	88	149	136	152	184	155	232	158	280	161
41	164	89	167	137	170	185	173	233	176	281	179
42	182	90	185	138	188	186	191	234	194	282	197
43	200	91	203	139	206	187	209	235	212	283	215
44	218	92	221	140	224	188	227	236	230	284	233
45	236	93	239	141	242	189	245	237	248	285	251
46	254	94	257	142	260	190	263	238	266	286	269
47	272	95	275	143	278	191	281	239	284	287	287

[0024] Aus der Tabelle 6 wird verständlich, daß, wenn die Eingabe-Bits spaltenweise in einer Matrix mit 16 Zeilen und 18 Spalten aufgeschrieben werden, man danach die Bits zeilenweise von oben in der korrekten Reihenfolge lesen kann. Ferner ist die genannte Spaltengröße 18 genau das Dreifache der Bitanzahl pro Träger oder mit anderen Worten genau die Größe, die zum Ausführen der zweiten Permutation benötigt wird. Man kann zeigen, dass genau dasselbe Prinzip auf die anderen Modi anwendbar ist. Es ist ausreichend, die Anzahl der Spalten entsprechend auf 12, 6 oder 3 für 16 QAM, QPSK oder BPSK zu reduzieren.

[0025] In [Fig. 4](#) ist das Prinzip der ersten Permutation zusammengefasst. Mit der beschriebenen Durchführungsweise werden zum Schreiben des Inhalts eines kompletten OFDM-Symbols bis zu 288 Zyklen benötigt.

[0026] Zusammenfassend kann man also hinsichtlich des Standes der Technik folgendes feststellen:

1) Die durch das System nach dem Stand der Technik eingeführte Verzögerung ist hoch, da zur Punktierung eine Bitanzahl benötigt wird, die doppelt so groß ist wie die Anzahl der Eingabebits (maximal 432 Zyklen/OFDM-Symbol), und da zur Verschachtelung alle Eingabe-Bits seriell gelesen werden müssen (maximal 288 Zyklen/OFDM-Symbol). Die Latenz ist ein kritischer Parameter für die Auslegung von Burst-Übertragungssystemen wie etwa WLAN.

2) Man benötigt FIFO-Elemente, um die Geschwindigkeitsunterschiede zwischen dem Kodierer 1 und dem ersten Punktierer P1 sowie zwischen dem ersten Punktierer P1 und dem zweiten Punktierer P2 auszugleichen. Ferner benötigt man FIFO-Elemente, um den Geschwindigkeitsunterschied zwischen der parallelen Eingabe des zweiten Punktierers P2 und der seriellen Eingabe des Verschachtelers 2 auszugleichen.

3) Das Taktsystem muss schnell genug sein, um alles innerhalb eines OFDM-Symbols abschließen zu können. Bei einer Implementation mit einer Taktfrequenz von 80 MHz sind zwei OFDM-Symbole, und folglich zwei an den Kodierer 1 gelieferte Bit-Bursts, $3,6 \mu\text{s} \cdot 80 \text{ MHz} = 288$ Taktzyklen voneinander entfernt (schlimmster Fall mit Short-Guard-Intervall bei HIPERLAN; typischer Fall: $4,0 \mu\text{s} \cdot 80 \text{ MHz} = 320$ Taktzyklen). Während dieser Zeit müssen die Eingabedaten gelesen und irgendwo gespeichert werden. Danach müssen sie in der richtigen Reihenfolge (Gleichung 1) ausgelesen werden, und es ist die zweite Permutation (Gleichung 2) durchzuführen. Die Punktierer P1, P2 und der Verschachteler 2 brauchen bis zu 432 Taktzyklen, um alle Eingabebits zu lesen. Daher ist es nicht möglich, alle Operationen innerhalb eines OFDM-Symbols zu vollziehen, weswegen man nach dem Stand der Technik sogenannte Pipelines für gerade und ungerade OFDM-Symbole schafft, indem man die Schaltung verdoppelt oder sogar noch weiter vervielfacht.

[0027] Weitere Verfahren zur Fehlererkennung und Fehlerkorrektur, insbesondere zu Verschachtelungsverfahren (Interleaving), sind auch in den im folgenden angegebenen Druckschriften offenbart. Die Druckschrift

„Sweeney, P.: Codierung zur Fehlererkennung und Fehlerkorrektur, München, Hanser, 1992, ISBN: 3-446-16439-1, S. 144–146“ beschreibt Verfahren zur Verschachtelung (Interleaving). Die Druckschrift EP 0 301 161 B1 beschreibt ein Verfahren zur Aufbereitung eines Faltungscodes zur Übertragung sowie dessen empfangsseitigen Rückwandlung sowie eine Anordnung hierzu. Die Druckschrift US 2001/0034868 A1 beschreibt einen Verschachteler-Aufbau für geparte bzw. zergliederte parallel verkettete Codes. Die Druckschrift DE 195 24 556 A1 beschreibt einen Faltungscodes sowie einen Faltungsdecoder zur Aufbereitung eines punktierten Faltungscodes. Die Druckschrift DE 100 48 872 A1 beschreibt ein Verfahren zur Entschachtelung eines blockweise verschachtelten Datensignals.

[0028] Der Erfindung liegt daher die Aufgabe zugrunde, elektronische Sendervorrichtungen mit einer Punktierereinrichtung und/oder einem Verschachteler, elektronische Empfängervorrichtungen mit einem Entschachteler und/oder einer Depunktierereinrichtung sowie ein Nachrichtenübertragungssystem mit einer Punktierereinrichtung und/oder einem Verschachteler und/oder einem Entschachteler und/oder einer Depunktierereinrichtung bereitzustellen, welche die oben mit Bezug auf den Stand der Technik erläuterten Geschwindigkeitsdifferenzprobleme bei der Datenverarbeitung überwinden.

[0029] Erfindungsgemäß wird diese Aufgabe gelöst durch eine Punktierereinrichtung zur Verwendung in einer elektronischen Sendervorrichtung nach Anspruch 1, durch einen Verschachteler zur Verwendung in einer elektronischen Sendervorrichtung nach Anspruch 10, durch einen Entschachteler zur Verwendung in einer elektronischen Empfängervorrichtung nach Anspruch 21, durch eine Depunktierereinrichtung zur Verwendung in einer elektronischen Empfängervorrichtung nach Anspruch 33 und durch ein Nachrichtenübertragungssystem nach Anspruch 40.

[0030] Die Überwindung der nach dem Stand der Technik vorhandenen Geschwindigkeitsunterschiede erfolgt bei den erfindungsgemäßen Vorrichtungen durch die Parallelisierung der Datenströme.

[0031] Unter dem Begriff "Nachrichtenübertragungssystem" ist ganz allgemein ein System zur Übertragung beliebiger Information, d. h. z. B. Sprache, Bilder, Daten usw., zu verstehen.

[0032] Vorteilhafte und bevorzugte Ausführungsformen der erfindungsgemäßen Punktierereinrichtung nach Anspruch 1 sind Gegenstand der Ansprüche 2 bis 9, 19 und 20. Vorteilhafte und bevorzugte Ausführungsformen des erfindungsgemäßen Verschachtelers nach Anspruch 10 sind Gegenstand der Ansprüche 11 bis 18. Vorteilhafte und bevorzugte Ausführungsformen des erfindungsgemäßen Entschachtelers nach Anspruch 21 sind Gegenstand der Ansprüche 22 bis 32. Vorteilhafte und bevorzugte Ausführungsformen der erfindungsgemäßen Depunktierereinrichtung nach Anspruch 33 sind Gegenstand der Ansprüche 34 bis 39. Vorteilhafte und bevorzugte Ausführungsformen des erfindungsgemäßen Nachrichtenübertragungssystems sind Gegenstand der Ansprüche 41 bis 44.

[0033] Ausführungsbeispiele der Erfindung werden nachfolgend anhand von Figuren erläutert. Es zeigt:

[0034] [Fig. 1](#) ein Ausführungsbeispiel einer erfindungsgemäßen Sendervorrichtung,

[0035] [Fig. 2](#) eine Sendervorrichtung nach dem Stand der Technik,

[0036] [Fig. 3](#) eine Empfängervorrichtung nach dem Stand der Technik,

[0037] [Fig. 4](#) das Prinzip der ersten Permutation,

[0038] [Fig. 5](#) ein Ausführungsschema für die Ausführung einer ersten Punktierung gemäß der Erfindung,

[0039] [Fig. 6](#) ein Ausführungsbeispiel der Schaltung eines ersten Punktierers gemäß der Erfindung,

[0040] [Fig. 7](#) das zur Schaltung von [Fig. 6](#) gehörende Zeitablauf-Diagramm,

[0041] [Fig. 8](#) ein Ausführungsschema für die Ausführung einer zweiten Punktierung gemäß der Erfindung,

[0042] [Fig. 9](#) ein Ausführungsbeispiel der Schaltung eines zweiten Punktierers gemäß der Erfindung,

[0043] [Fig. 10](#) das zur Schaltung von [Fig. 9](#) gehörende Zeitablauf-Diagramm,

- [0044] [Fig. 11](#) ein Ausführungsbeispiel einer erfindungsgemäßen Empfängervorrichtung,
- [0045] [Fig. 12](#) ein Beispiel für eine Depunktierungsfunktion in Übereinstimmung mit der Erfindung für die Koderate $3/4$,
- [0046] [Fig. 13a](#) ein Ausführungsbeispiel der Schaltung eines ersten Depunktierers gemäß der Erfindung,
- [0047] [Fig. 13b](#) ein Ausführungsbeispiel der Schaltung eines zweiten Depunktierers gemäß der Erfindung,
- [0048] [Fig. 14](#) Beispiele für die erste Depunktierungsfunktion in Übereinstimmung mit der Erfindung für verschiedene Koderaten,
- [0049] [Fig. 15](#) eine modifizierte Bit-Abbildung,
- [0050] [Fig. 16](#) ein Ausführungsbeispiel eines erfindungsgemäßen Verschachtelers mit Registern,
- [0051] [Fig. 17](#) ein Umordnungsschema der Speicherelemente im Verschachteler,
- [0052] [Fig. 18](#) schematisch eine Schreibphase der ersten Permutation im Verschachteler für ein Ausführungsbeispiel des Verschachtelers mit RAM,
- [0053] [Fig. 19](#) schematisch eine Lesephase der ersten Permutation im Verschachteler für ein Ausführungsbeispiel des Verschachtelers mit RAM,
- [0054] [Fig. 20](#) schematisch ein Ausführungsbeispiel der Schreibphase bei der Invertierung der ersten Permutation im Entschachteler und
- [0055] [Fig. 21](#) schematisch ein Ausführungsbeispiel der Lesephase bei der Invertierung der ersten Permutation im Entschachteler.
- [0056] Es ist möglich, die in den oben genannten Standards definierten Gleichungen so umzuformulieren, daß der gesamte Datenpfad in zwei parallelen Leitungen abgewickelt werden kann. [Fig. 1](#) zeigt ein entsprechendes Blockschaltbild für die Senderseite.
- [0057] Aus [Fig. 5](#) ist ersichtlich, daß die vom ersten Punktierer P1 durchzuführende erste Punktierung in paralleler Weise realisiert werden kann. Wenn man abwechselnd die parallelen Ausgaben Out_X und Out_Y liest, erhält man die ursprüngliche Sequenz. Der einzige Unterschied besteht darin, daß es in diesem Fall auch einen leeren Platz gibt, der bei der Ausgabesequenz von [Fig. 5](#) zeitlich gesehen genau in der Mitte liegt. Der dem ersten Punktierer nachfolgende Hardwareblock, d. h. der zweite Punktierer P2, muß mit einem Signal, welches im Folgenden "data_valid-Signal" genannt wird, darüber informiert werden, daß er den leeren Platz nicht beachten soll.
- [0058] Im schnellsten Modus mit 222 Eingabe-Bits erzeugt der Kodierer $1\ 2 \times 222$ Bits, und die Ausgabelänge bleibt 2×222 (mit 2×6 Lücken). Eine Möglichkeit zur hardwaremäßigen Realisierung der beschriebenen Funktionsweise ist die in [Fig. 6](#) dargestellte Schaltung.
- [0059] Bei der Schaltung von [Fig. 6](#) ist der erste Dateneingang IN_X des dort dargestellten ersten Punktierers P1 über ein erstes 1-Schritt-Verzögerungsregister D mit dem ersten Datenausgang Out_X des ersten Punktierers P1 verbunden (obere Datenleitung in [Fig. 6](#)). Die untere Datenleitung von [Fig. 6](#) zeigt die Verknüpfung des zweiten Dateneingangs IN_Y des ersten Punktierers P1 mit seinem zweiten Datenausgang Out_Y. Wesentliches Element in dieser Verbindung ist ein Multiplexer MUX, auf dessen ersten Eingang die über den zweiten Dateneingang IN_Y des ersten Punktierers P1 in den ersten Punktierer P1 einlaufenden Daten nach Durchlaufen eines 1-Schritt-Verzögerungsregisters D bereitgestellt werden. Parallel dazu werden die über den zweiten Dateneingang IN_Y des ersten Punktierers P1 in den ersten Punktierer P1 einlaufenden Daten auch unmittelbar auf einen zweiten Eingang des Multiplexers MUX geführt.
- [0060] Der Multiplexer MUX weist einen Ausgang auf, der elektrisch über ein weiteres 1-Schritt-Verzögerungsregister D mit dem zweiten Datenausgang Out_Y des ersten Punktierers P1 verbunden ist.
- [0061] Wie leicht einzusehen ist, kann bei dem Schaltungsbeispiel gemäß [Fig. 6](#) mittels geeigneter Einstel-

lung des Multiplexers MUX und geeigneter Einrichtung der 1-Schritt-Verzögerungsregister D ausgewählt werden, wann das Bitpaar entfernt werden soll. [Fig. 7](#) zeigt das entsprechende Zeitablauf-Diagramm. Die Datenbits eines OFDM-Symbols (222) werden in einem einzigen Burst an den Faltungskodierer **1** übertragen. Dieser erzeugt auf zwei parallelen Leitungen einen Burst der gleichen Länge. Der erste Punktierer P1 ordnet die beiden Kodiererausgaben um und teilt, indem er das data_valid-Signal auf LOW setzt, dem zweiten Punktierer P2 die Positionen der 6 Lücken mit.

[0062] FIFO-Speicherelemente werden bei derartigen erfindungsgemäßen Sendervorrichtungen nicht mehr benötigt, weil der Kodierer **1** und die den ersten Punktierer P1 sowie den zweiten Punktierer P2 aufweisende Punktierereinheit P1, P2 mit gleicher Geschwindigkeit laufen. Außerdem wird die Zeit zur Durchführung der senderseitigen Bearbeitungsoperation stark verringert (222 Zyklen). Daher kann man bei Anwendung der Erfindung eine einzelne Struktur mit einer Taktfrequenz von 62 MHz anstelle von 120 nach dem Stand der Technik verwenden, benötigt also keine "Pipelines" mehr.

[0063] Man kann nun das gleiche Prinzip auf den zweiten Punktierer P2 anwenden. In [Fig. 8](#) sind alle möglichen Punktierungsschemata dargestellt, allerdings fehlt die Koderate 1/2, weil diese bedeutet "tue nichts". Wieder beträgt die von dieser Einheit, also vom zweiten Punktierer P2, eingeführte Latenz einen Taktzyklus, während die Zeit, die zur Durchführung der Operation benötigt wird, mit der Anzahl der Eingabedaten-Bits übereinstimmt (maximal 222).

[0064] [Fig. 9](#) zeigt ein Beispiel für eine erfindungsgemäße Implementierung des zweiten Punktierers P2. Bei der in [Fig. 9](#) dargestellten Schaltung weist der zweite Punktierer P2 drei Multiplexer MUX auf, von denen wiederum jeder zwei Eingänge und einen Ausgang aufweist und darüber hinaus an eine Steuerleitung mux_0, mux_1 angeschlossen ist.

[0065] Der erste Dateneingang IN_X des zweiten Punktierers P2 ist elektrisch unmittelbar sowohl mit dem ersten Eingang des ersten Multiplexers des zweiten Punktierers P2 als auch mit dem ersten Eingang des zweiten Multiplexers des zweiten Punktierers P2 verbunden. Der zweite Dateneingang IN_Y des zweiten Punktierers P2 ist elektrisch unmittelbar sowohl mit dem zweiten Eingang des ersten Multiplexers des zweiten Punktierers P2 als auch mit dem zweiten Eingang des zweiten Multiplexers des zweiten Punktierers P2 verbunden.

[0066] Der Ausgang des ersten Multiplexers des zweiten Punktierers P2 ist elektrisch unmittelbar mit dem ersten Eingang des dritten Multiplexers des zweiten Punktierers P2 verbunden. Parallel dazu ist der Ausgang des ersten Multiplexers des zweiten Punktierers P2 auch elektrisch über ein 1-Schritt-Verzögerungsregister D mit dem zweiten Eingang des dritten Multiplexers des zweiten Punktierers P2 verbunden.

[0067] Der Ausgang des dritten Multiplexers des zweiten Punktierers P2 ist elektrisch über ein 1-Schritt-Verzögerungsregister D an den ersten Datenausgang Out_X des zweiten Punktierers P2 angeschlossen. Der Ausgang des zweiten Multiplexers des zweiten Punktierers P2 ist elektrisch über ein weiteres 1-Schritt-Verzögerungsregister D an den zweiten Datenausgang Out_Y des zweiten Punktierers P2 angeschlossen.

[0068] Aus [Fig. 9](#) wird deutlich, daß mit einer geeigneten Steuerung der Multiplexer MUX und geeigneter Einrichtung der 1-Schritt-Verzögerungsregister D leicht die gewünschten Ausgabesequenzen erzielt werden können.

[0069] Bei dem in [Fig. 10](#) dargestellten Zeitdiagramm für das zweite Punktieren wurde der erste Teil der Ausgabe gemäß [Fig. 7](#) ausgedehnt. Es ist in [Fig. 10](#) die Ausgabe des zweiten Punktierers P2 für die 3/4-Rate dargestellt. Die bereits durch den ersten Punktierer P1 eingeführten Lücken verbleiben in der gleichen Position, jedoch erkennt man darüber hinaus die neu eingeführten Lücken. Das data_valid-Signal ist für beide Arten von Lücken LOW gesetzt.

[0070] Schließlich muss auch der Verschachteler **2** derart ausgelegt werden, daß er in der Lage ist, zwei parallele Eingaben anstelle von einer zu verarbeiten. Ehe dies jedoch weiter unten ausführlich beschrieben wird, soll zunächst die neue Art und Weise der Depunktierung auf der Empfängerseite behandelt werden.

[0071] Die erfindungsgemäße Art und Weise der Depunktierung auf der Empfängerseite funktioniert prinzipiell genauso, wie die erfindungsgemäße Art und Weise der Punktierung auf der Senderseite. [Fig. 11](#) zeigt ein entsprechendes Blockschaltbild mit Entschachteler **3**, erstem Depunktierer P2', zweitem Depunktierer P1' und Viterbi-Dekodierer **4**. In [Fig. 12](#) ist die Depunktierungsfunktion für die Koderate 3/4 dargestellt, wobei anstelle der fehlenden Bits Softnullen eingefügt werden. Hier wird vorausgesetzt, daß der Entschachteler **3** schon in

der Lage ist, Lücken am Ende einer Depunktierungsgruppe einzufügen. Der Entschachteler **3** vollführt also bereits die FIFO-Funktion, um die verschiedenen Eingabe/Ausgabe-Geschwindigkeiten einander anzupassen, und es wird kein zusätzlicher separater Speicher benötigt. Für genauere diesbezügliche Ausführungen sei auf die weiter unten dargestellten Erläuterungen zur erfindungsgemäßen Funktionsweise des Entschachtelers **3** verwiesen.

[0072] Wenn auch der erste Punktierer P1 zum Einsatz kommt, muß der Entschachteler **3** dafür sorgen, daß auch eine Lücke am Ende einer ersten Punktierungsgruppe gelassen wird. Der erste Depunktierer P2' füllt zunächst die Lücken, die zur zweiten Punktierungsgruppe gehören, und schließlich füllt der zweite Depunktierer P1' die verbleibenden Lücken. Alle Bursts in jedem Abschnitt haben eine Länge, die gleich der Anzahl unkodierter Bits ist. Daher beträgt die Latenz für den ersten Depunktierer P2' einen Zyklus und einen weiteren Zyklus für den zweiten Depunktierer P1'.

[0073] In [Fig. 12](#) wurde der Fall der 3/4-Rate dargestellt, jedoch kann das Prinzip auf alle anderen Koderaten ausgedehnt werden. Dies ist für den ersten Depunktierer P2' in [Fig. 14](#) gezeigt.

[0074] Ein Beispiel für eine hardwaremäßige Ausführung des ersten Depunktierers P2' ist in [Fig. 13a](#) dargestellt, und [Fig. 13b](#) zeigt ein Beispiel für eine hardwaremäßige Ausführung des zweiten Depunktierers P1'. Bei der Schaltung gemäß [Fig. 13a/13b](#) wird die Eingabe/Ausgabe durch N Soft-Bits für den nachfolgenden Viterbi-Dekodierer **4** repräsentiert.

[0075] Bei dem Ausführungsbeispiel des ersten Depunktierers P2' gemäß [Fig. 13a](#) weist der erste Depunktierer P2' einen ersten und einen zweiten Multiplexer MUX mit jeweils zwei Eingängen und einem Ausgang sowie einen dritten Multiplexer MUX mit vier Eingängen und einem Ausgang auf.

[0076] Jeweils ein 1-Schritt-Verzögerungsregister D ist geschaltet zwischen den Ausgang des ersten Multiplexers und einen Eingang des zweiten Multiplexers, zwischen den Ausgang des zweiten Multiplexers und einen ersten Datenausgang Out_X des ersten Depunktierers P2', zwischen den Ausgang des dritten Multiplexers und einen zweiten Datenausgang Out_Y des ersten Depunktierers P2' und zwischen einen ersten Dateneingang IN_Y des ersten Depunktierers P2' und einen Eingang des dritten Multiplexers.

[0077] Der erste Dateneingang IN_Y des ersten Depunktierers P2' ist elektrisch unmittelbar ferner an einen Eingang des ersten Multiplexers und an einen weiteren Eingang des dritten Multiplexers angeschlossen. Der zweite Dateneingang IN_X des ersten Depunktierers P2' ist elektrisch unmittelbar an den weiteren Eingang des zweiten Multiplexers und an den dritten Eingang des dritten Multiplexers angeschlossen. Der jeweils verbleibende Eingang des ersten Multiplexers und des dritten Multiplexers ist an eine Leitung angeschlossen, auf der Soft-Nullen bereitgestellt werden.

[0078] Bei dem Ausführungsbeispiel des zweiten Depunktierers P1' gemäß [Fig. 13b](#) weist der zweite Depunktierer P1' drei Multiplexer MUX mit jeweils zwei Eingängen und einem Ausgang auf. Jeweils ein 1-Schritt-Verzögerungsregister D ist geschaltet zwischen den Ausgang des ersten Multiplexers und einen Eingang des zweiten Multiplexers, zwischen den Ausgang des zweiten Multiplexers und den ersten Datenausgang Out_X des zweiten Depunktierers P1' und zwischen den Ausgang des dritten Multiplexers und den zweiten Datenausgang Out_Y des zweiten Depunktierers P1'. Der erste Dateneingang IN_X des zweiten Depunktierers P1' ist elektrisch unmittelbar an einen Eingang des ersten Multiplexers und an den weiteren Eingang des zweiten Multiplexers angeschlossen. Der zweite Dateneingang IN_Y des zweiten Depunktierers P1' ist elektrisch unmittelbar an einen Eingang des dritten Multiplexers angeschlossen. Der jeweils verbleibende Eingang des ersten und des dritten Multiplexers ist an eine Leitung angeschlossen, auf der Soft-Nullen bereitgestellt werden.

[0079] Aus den [Fig. 13a](#) und [13b](#) wird deutlich, daß mit einer geeigneten Steuerung der Multiplexer MUX und einer geeigneten Einrichtung der 1-Schritt-Verzögerungsregister D leicht die gewünschten Ausgabesequenzen erzielt werden können.

[0080] Wenden wir uns nun einer genaueren Beschreibung von Ausführungsbeispielen der Verschachteler **2** bzw. Entschachteler **3** in den erfindungsgemäßen Sender-Empfänger-Vorrichtungen zu.

[0081] Bzgl. der senderseitigen Stellung des Verschachtelers **2** sei auf die [Fig. 1](#) verwiesen. Ferner sei daran erinnert, daß der zweite Punktierer P2 in der Lage ist, an seinem ersten Datenausgang Out_X das gerade Bit gemäß Gleichung 1 und an seinem zweiten Datenausgang Out_Y die ungeraden Bits gemäß Gleichung 1 be-

reitzustellen.

[0082] Zur Erläuterung erfindungsgemäßer Verschachteler-Ausführungsbeispiele werden im Folgenden zwei neue mögliche Implementationen des ersten Permutationsschemas dargestellt. Die erste Implementation wird mit Registern verwirklicht, und das Verständnis der ersten Implementation ist nützlich für das Verständnis der weiter unten erläuterten zweiten Verwirklichungsmöglichkeit mittels RAM.

[0083] Zunächst wird die Spalten-Träger-Abbildung von [Fig. 4](#) modifiziert, so daß sich die neue Abbildung gemäß [Fig. 15](#) ergibt. Auf diese Weise ergibt sich, daß die drei Gruppen von sechs Reihen unabhängig von dem ausgewählten Übertragungsmodus immer zu den gleichen Trägern gehören.

[0084] [Fig. 16](#) zeigt als Beispiel eine Verschachteler-Ausführung mit Registern. Dieses Ausführungsbeispiel gemäß [Fig. 16](#) beinhaltet zwei Reihen von 8 Bit-Schieberegistern. Der zweite Punktierer P2 liefert dem Verschachteler **2** gerade Bits (Out_X) und ungerade Bits (Out_Y), die innerhalb der geraden/ungeraden Registergruppe verschoben werden. Nach 2×8 Eingaben sind die Register voll. Man kann nun eine vollständige Spalte der Matrix des neben den Schieberegistern vorhandenen Matrixregisters, das im vorliegenden Ausführungsbeispiel ein 16×18 -Matrixregister ist, schreiben und dann wieder mit dem Füllen der geraden/ungeraden Schieberegister beginnen. Abhängig vom gewählten Modus werden die Spalten (0, 12, 6)...(0, 1, 2..., 17) jeweils entsprechend für BPSK...64 QAM geschrieben. Für den 64 QAM-Modus benötigt die Phase des Schreibens in das Matrixregister 18 (Anzahl der Spalten) \times 8 (benötigte Zeit zum Füllen der Schieberegister) = 144 Zyklen. Folglich wird diese Zeit, verglichen mit der Zeit, die bei der in den o. g. Standards spezifizierten seriellen Implementation benötigt wird, halbiert.

[0085] Für die Lesephase müssen nun die Matrixzeilen gelesen werden. Eine Zeile enthält immer die Bits, die zum Abbilden der drei QAM-Träger benötigt werden. Man kann die zweite Permutation (Gleichung 2) auf die komplette Zeile mit einem Mal anwenden oder seriell dreimal die gleiche Permutation auf den drei Trägern ausführen. Dementsprechend dauert die Lesephase entweder 16 oder 48 Zyklen. Die Gesamtzeit, die für das Verschachteln im letztgenannten Fall benötigt wird, beträgt $144 + 48 = 192$ Zyklen. Daher muß nur ein einziger Verschachteler **2** mit einer Taktfrequenz von größer als 53,3 MHz benutzt werden. In dem erwähnten Beispiel mit einer Taktfrequenz von 80 MHz sind noch 96 Zyklen frei. Diese freien Zyklen können für den Fall benutzt werden, daß die Punktierereinrichtung P1, P2 die Eingabebits mit Lücken dazwischen bereitstellt. Dies ist genau der Fall, wie er bereits mit Bezug auf das obige Ausführungsbeispiel der erfindungsgemäßen Punktierereinrichtung P1, P2 beschrieben wurde, bei dem also die Eingabe an den Verschachteler **2** über 222 Zyklen verteilt ist. Die gesamte Verschachtelungszeit wird dann 270 Zyklen lang, wobei die minimale Taktfrequenz, bei der der Pipelines vermieden werden können, 75 MHz beträgt.

[0086] Ausgehend von dem eben beschriebenen Ausführungsbeispiel des Verschachtelers mit Registern kann man die Speicherelemente so umordnen, daß es möglich ist, eine geeignetere Konfiguration für eine RAM-Implementation des Verschachtelers **2** zu erhalten. Bezogen auf die Darstellung von [Fig. 16](#) sei bemerkt, daß der linke Block des Matrixregisters zuerst geschrieben wird, der mittlere Block als zweiter und schließlich der rechte Block als letzter. Es werden diese drei Blöcke nun in der genannten Reihenfolge spaltenweise geschrieben (siehe [Fig. 17](#)).

[0087] Um die beiden Bits, die von der Punktierereinheit P1, P2 kommen, zu schreiben, müßte man auf zwei Zeilen (oder eine Spalte) in einem Zyklus zugreifen. Bei einem Standard-RAM möchte man jedoch nur auf eine Zeile pro Zyklus zugreifen. Deshalb spalten wir weiter jeden Block in zwei Spalten auf, eine mit den geraden Zeilen und die zweite mit den ungeraden Zeilen (siehe [Fig. 17](#)). So liegen das gerade und das ungerade Bit, die von der Punktierereinheit P1, P2 kommen, jetzt auf der gleichen Zeile, und sie können simultan in einem einzigen Zyklus geschrieben werden. Schließlich erhält man einen RAM-Block mit 24 Zeilen und 12 Spalten, wie in [Fig. 18](#) gezeigt. Im Folgenden wird der 64 QAM-Fall erläutert. Darauf aufbauend ist die Erweiterung auf die anderen Modi dann trivial.

[0088] Man muß in der Lage sein, genau 2 Bits in einer Zeile von 12 zu schreiben (z. B. RAM mit Einzelbit-Schreiben). Zunächst konzentrieren wir uns auf den ersten Block von 8 Zeilen (das ist der linke Block in der Register-Implementation von [Fig. 16](#)). Wir schreiben zuerst die Zeilen von 0 bis 7 und immer die Spalten 0 und 6. Auf diese Weise werden die ersten 2×8 Bits geschrieben. Dann schreiben wir wieder die Zeilen 0 bis 7, jedoch dieses Mal immer die Spalten 1 und 7 usw. bis zu den Bits 94, 95. Jetzt wechseln wir den Zeilenblock und betrachten die Zeilen von 8 bis 15 und wiederholen alles bis zum Bit 191. Dann wird zum letzten Mal der Block gewechselt, und es werden die Zeilen von 16 bis 23 bis zum letzten Bit bearbeitet (siehe [Fig. 18](#)).

[0089] Nun muß man die Bits in der richtigen Reihenfolge lesen und so für die zweite Permutation vorbereiten. Es ist leicht einzusehen, daß die Gruppen von 6 Bits aus dem Speicher in der Reihenfolge, wie sie in [Fig. 19](#) dargestellt ist, herausgeholt werden müssen.

[0090] Folglich muß man die Zeilen mit folgenden Adressen lesen: 0, 8, 16, 0, 8, 16, 1, 9, 17, 1, 9, 17, ..., 23. Dreimal extrahieren wir von den 12 Bits die 6 MSB (MSB = Most Significant Bits, links stehende Bits) und wenden das gleiche zweite Permutationsschema an. Beim nächsten Mal extrahieren wir die 6 LSB (LSB = Least Significant Bits, rechts stehende Bits) und wenden das darauffolgende zweite Permutationsschema an.

[0091] Wie oben gesagt, bezieht sich das erläuterte Beispiel auf den 64 QAM-Fall. Es ist einfach zu prüfen, daß die Permutation die gestellten Erfordernisse erfüllt. Ferner ist es einfach, das erläuterte System auf BPSK, QPSK und 16 QAM zu übertragen. Man muß nur die Anzahl der Spalten während der Schreibphase jeweils entsprechend auf die Spalten (0, 6), (0...1, 6...7) bzw. (0...3, 6...9) verringern.

[0092] Zusammenfassend gesagt besteht der Vorteil der RAM-Ausführungsform des Verschachtelers **2** gegenüber der Register-Ausführungsform des Verschachtelers **2** darin, daß bei der RAM-Ausführungsform keine Zwischenspeicherung der Datenbits in Schieberegistern erfolgt, sondern die Bitpaare direkt in den RAM auf die richtigen Adressen geschrieben werden.

[0093] Wie bereits oben mit Bezug auf die Depunktierereinheit ($P2'$, $P1'$) beschrieben, wird auch empfängerseitig die Standard-Lösung zu einer komplett parallelen Lösung umgestaltet, wie es in dem Blockschaltbild von [Fig. 11](#) dargestellt ist. Die Demapper-Einheit, welche dem Entschachteler **3** vorgeschaltet ist, versorgt den Entschachteler **3** mit einem Träger pro Zyklus. Jeder Träger enthält 1, 2, 4 oder 6 Softbits entsprechend dem ausgewählten Modulationsmodus (Abbildung wie in [Fig. 15](#)). Ein Softbit ist gleich dem bestimmten Bit plus $N - 1$ Zuverlässigkeitsbits, die bis zum Viterbi-Dekodierer **4**, dessen Ausgabe ein einzelnes bestimmtes Bit ist, zusammen verarbeitet werden müssen. Daher ist in diesem Fall das Basis-Datenelement nicht mehr das Bit, sondern eine Gruppe von N Bits, wobei N typischerweise gleich 4 ist. Auf der Empfängerseite ist die RAM-Lösung die bevorzugte Ausführungsform, weil die erforderlichen Speicherelemente um einen Faktor N anwachsen. Auch hier benötigt man bei der RAM-Lösung keine Zwischenspeicherung der (Ausgabe)bits in (Ausgabe-)Schieberegistern, sondern das Auslesen der Bits aus dem RAM erfolgt paarweise direkt in die Depunktierereinrichtung ($P2'$, $P1'$).

[0094] Auf der Empfängerseite müssen die zweite und die erste Permutation invers ausgeführt werden.

[0095] Nach der Durchführung der Inversen der zweiten Permutation schreitet jeder Träger (Gruppe von 1, 2, 4 oder 6 Softbits entsprechend den Modi BPSK, QPSK, 16 QAM oder 64 QAM) weiter zur Durchführung der Inversen der ersten Permutation. Hier kann wiederum ein RAM mit 24 Zeilen zu $12 \times N$ Bits verwendet werden. Die Schreibphase folgt dem gleichen Schema wie die Lese-Phase auf der Senderseite und ist in [Fig. 20](#) dargestellt. Die Lese-Phase ist in [Fig. 21](#) gezeigt und folgt dem gleichen Schema wie die Schreibphase auf der Senderseite.

[0096] Die Gesamtzeit, die benötigt wird, um das Entschachteln eines OFDM-Symbols durchzuführen, beträgt 48 Zyklen für die Schreibphase bei allen Übertragungsmodi + 144 Zyklen für die Lese-Phase bei 64 QAM. Um den Betrieb der dem Entschachteler **3** nachgeschalteten Depunktierereinheit ($P2'$, $P1'$) zu erleichtern, können während der Lese-Phase zwischen den Daten einige Lücken gelassen werden (222 Zyklen). Für diesbezügliche weitere Details sei auf die obigen Ausführungen zur Depunktierereinrichtung ($P2'$, $P1'$) verwiesen.

[0097] Genauso einfach, wie es vorstehend für die RAM-Implementation erläutert wurde, ist es, empfängerseitig eine Invertierung der Verschachteler-Register-Implementation als erfindungsgemäßes Ausführungsbeispiel zu implementieren. Der Dateneingabevorgang, wie er oben mit Bezug auf das Register-Beispiel des Verschachtelers **2** beschrieben wurde, muß nur in einen Datenausgabevorgang umgekehrt werden. Empfängerseitig wird dann also eine Softbit-Spalte aus dem Matrixregister kammartig verzahnt in zwei Softbit-Schieberegister eingelesen, d. h. zwei benachbarte Softbits werden jeweils in ein anderes der beiden Softbit-Schieberegister eingelesen. Aus den beiden Softbit-Schieberegistern erfolgt die jeweils serielle, jedoch aufgrund des Vorhandenseins zweier Softbit-Schieberegister insgesamt parallele Softbitausgabe in die Depunktierereinrichtung ($P2'$, $P1'$). Sobald die beiden Softbit-Schieberegister leer sind, wird in gleicher Weise wie bei der ersten aus dem Matrixregister ausgelesenen Spalte eine weitere Spalte aus dem Matrixregister in beide Softbit-Schieberegister ausgelesen, dann werden beide Softbit-Schieberegister gleichzeitig geleert und so fort.

[0098] Hardwaremäßig werden bei einem Ausführungsbeispiel des Entschachtelers **3** z. B. das Matrixregister

als $16 \times (18 \times N)$ -Matrixregister und die beiden Softbit-Schieberegister als 8-Softbit-Schieberegister mit einer Wortbreite von N ausgeführt.

[0099] Zusammenfassend kann man die mit der erfindungsgemäßen Lösung erzielten Vorteile wie folgt zusammenfassen:

- 1) Verzögerungsreduktion beim Sender und beim Empfänger. Die Latenz oder Verzögerung ist gewöhnlicherweise ein besonders kritischer Parameter für die Auslegung eines Burst-Übertragungssystems. Die in der Punktierereinrichtung/Depunktierereinrichtung P1, P2, P2', P1' bzw. im Verschachtelter/Entschachtelter **2, 3** eingesparte Latenz kann folglich nutzbringend in anderen Teilen des Burst-Übertragungssystems, z. B. beim Equalizer oder bei der Synchronisation, eingeplant werden, um dort leistungsfähigere Algorithmen implementieren zu können.
- 2) Verringerung des Platz- bzw. Raumbedarfs, da kein FIFO mehr benötigt wird.
- 3) Damit weiter verbunden sind außerdem geringe Taktfrequenzen. Realisiert man ein WLAN-System mit einem Takt von 80 MHz, d. h. dem Vierfachen der OFDM-Abtastfrequenz, auf erfindungsgemäße Weise, so muß der Datenpfad nicht mehr, wie nach dem Stand der Technik erforderlich, in ungerade/gerade OFDM-Symbole verdoppelt werden.

[0100] Verwendet man senderseitig eines der oben dargestellten Ausführungsbeispiele für die erfindungsgemäße Sendervorrichtung und empfängerseitig eines der oben dargestellten Ausführungsbeispiele für die erfindungsgemäße Empfängervorrichtung, so erhält man ein Ausführungsbeispiel für ein erfindungsgemäßes Nachrichtenübertragungssystem, das z. B. ein WLAN sein kann.

Patentansprüche

1. Punktierereinrichtung zur Verwendung in einer elektronischen Sendervorrichtung, wobei die Punktierereinrichtung

- einen ersten und einen zweiten Datenausgang aufweist,
- so ausgelegt ist, daß sie ihren Ausgabedatenstrom im Wesentlichen gleichmäßig parallel auf ihre beiden genannten Datenausgänge verteilt,
- leere Plätze in ihrem Ausgabedatenstrom vorsieht, so dass die Anzahl der Bits des Eingangsdatenstroms der Anzahl der Bits des Ausgangsdatenstroms einschließlich der leeren Plätze entspricht, und
- zusätzlich zu ihrem parallelen Ausgabedatenstrom ein Signal (data_valid) ausgibt, welches leere Plätze im parallelen Ausgabedatenstrom der Punktierereinrichtung anzeigt.

2. Punktierereinrichtung nach Anspruch 1, dadurch gekennzeichnet, daß die Punktierereinrichtung aus genau einem Punktierer (P2) besteht.

3. Punktierereinrichtung nach Anspruch 1, dadurch gekennzeichnet, daß die Punktierereinrichtung einen ersten Punktierer (P1) und einen in Datenstromrichtung hinter dem ersten Punktierer (P1) angeordneten zweiten Punktierer (P2) aufweist.

4. Punktierereinrichtung nach Anspruch 3, dadurch gekennzeichnet, daß

- der erste Punktierer (P1) einen ersten und einen zweiten Datenausgang aufweist und so ausgelegt ist, daß er seinen Ausgabedatenstrom im Wesentlichen gleichmäßig auf seine beiden Datenausgänge verteilt, und
- der zweite Punktierer (P2) einen ersten und einen zweiten Dateneingang aufweist, wobei der erste Dateneingang des zweiten Punktierers (P2) elektrisch direkt oder indirekt mit dem ersten Datenausgang des ersten Punktierers (P1) und der zweite Dateneingang des zweiten Punktierers (P2) elektrisch direkt oder indirekt mit dem ersten Datenausgang des ersten Punktierers (P1) verbunden ist.

5. Punktierereinrichtung nach Anspruch 4, dadurch gekennzeichnet, daß

- der erste Punktierer (P1) so ausgelegt ist, daß er zusätzlich zu seinem parallelen Ausgabedatenstrom dem zweiten Punktierer (P2) ein Signal (data_valid) übermittelt, welches den zweiten Punktierer (P2) über leere Plätze im parallelen Ausgabedatenstrom des ersten Punktierers (P1) informiert und
- der zweite Punktierer (P2) so ausgelegt ist, daß er mit Hilfe des genannten, vom ersten Punktierer (P1) zusätzlich übertragenen Signals (data_valid) die leeren Plätze im vom ersten Punktierer (P1) kommenden parallelen Eingabedatenstrom erkennt und bei der weiteren Datenverarbeitung nicht beachtet.

6. Punktierereinrichtung nach Anspruch 4 oder Anspruch 5, dadurch gekennzeichnet, daß der erste Punktierer (P1) einen ersten Dateneingang (IN_X) und einen zweiten Dateneingang (IN_Y) aufweist und so ausgelegt ist, daß

- zwischen den ersten Dateneingang (IN_X) und den ersten Datenausgang (Out_X) ein 1-Schritt-Verzögerungsregister (D) geschaltet ist,
- der zweite Dateneingang (IN_Y) über ein 1-Schritt-Verzögerungsregister (D) elektrisch mit einem ersten Eingang eines Multiplexers (MUX) und parallel dazu unmittelbar elektrisch mit einem zweiten Eingang eines Multiplexers (MUX) verbunden ist und
- der Multiplexer (MUX) einen Ausgang aufweist, welcher über ein weiteres 1-Schritt-Verzögerungsregister (D) elektrisch mit dem zweiten Datenausgang (Out_Y) des ersten Punktierers (P1) verbunden ist.

7. Punktierereinrichtung nach einem der Ansprüche 3 bis 6, dadurch gekennzeichnet, daß der zweite Punktierer (P2) zwei Datenausgänge aufweist.

8. Punktierereinrichtung nach Anspruch 7, dadurch gekennzeichnet, daß die beiden Datenausgänge des zweiten Punktierers (P2) gleichzeitig die beiden Datenausgänge der Punktierereinrichtung sind.

9. Punktierereinrichtung nach Anspruch 7 oder Anspruch 8, dadurch gekennzeichnet, daß
- der zweite Punktierer (P2) drei Multiplexer (MUX) aufweist, die jeweils zwei Eingänge und einen Ausgang aufweisen,
 - der erste Dateneingang (IN_X) des zweiten Punktierers (P2) elektrisch sowohl mit dem ersten Eingang des ersten Multiplexers des zweiten Punktierers (P2) als auch mit dem ersten Eingang des zweiten Multiplexers des zweiten Punktierers (P2) unmittelbar verbunden ist,
 - der zweite Dateneingang (IN_Y) des zweiten Punktierers (P2) elektrisch sowohl mit dem zweiten Eingang des ersten Multiplexers des zweiten Punktierers (P2) als auch mit dem zweiten Eingang des zweiten Multiplexers des zweiten Punktierers (P2) unmittelbar verbunden ist,
 - der Ausgang des ersten Multiplexers des zweiten Punktierers (P2) elektrisch unmittelbar mit dem ersten Eingang des dritten Multiplexers des zweiten Punktierers (P2) verbunden ist,
 - der Ausgang des ersten Multiplexers des zweiten Punktierers (P2) elektrisch über ein 1-Schritt-Verzögerungsregister (D) an den zweiten Eingang des dritten Multiplexers des zweiten Punktierers (P2) angeschlossen ist,
 - der Ausgang des dritten Multiplexers des zweiten Punktierers (P2) über ein 1-Schritt-Verzögerungsregister (D) elektrisch mit dem ersten Datenausgang (Out_X) des zweiten Punktierers (P2) verbunden ist und
 - der Ausgang des zweiten Multiplexers des zweiten Punktierers (P2) elektrisch über ein weiteres 1-Schritt-Verzögerungsregister (D) an den zweiten Datenausgang (Out_Y) des zweiten Punktierers (P2) angeschlossen ist.

10. Verschachteler (2) zur Verwendung in einer elektronischen Sendervorrichtung, wobei der Verschachteler (2)

- zwei parallele Dateneingänge aufweist und
- so ausgelegt ist, daß er auf beiden Dateneingängen parallel einlaufende Datenströme verarbeiten kann,
- ein Blockinterleaver ist und
- einen einzigen zusammenhängenden Speicherblock mit Zeilen und Spalten umfasst.

11. Verschachteler (2) nach Anspruch 10, dadurch gekennzeichnet, daß der Verschachteler (2) ein $n \times m$ -Verschachteler ist, wobei n und m natürliche Zahlen sind.

12. Verschachteler (2) nach Anspruch 11, dadurch gekennzeichnet, daß der Verschachteler (2) ein elektrisch direkt oder indirekt an seinen ersten Dateneingang angeschlossenes erstes Schieberegister und ein elektrisch direkt oder indirekt an seinen zweiten Dateneingang angeschlossenes zweites Schieberegister aufweist.

13. Verschachteler (2) nach Anspruch 12, dadurch gekennzeichnet, daß beide Schieberegister $\frac{1}{2}n$ -Bit-Schieberegister sind.

14. Verschachteler (2) nach Anspruch 12 oder 13, dadurch gekennzeichnet, daß der Verschachteler (2) ein Matrixregister aufweist.

15. Verschachteler (2) nach Anspruch 14, dadurch gekennzeichnet, daß das Matrixregister ein 16×18 -Matrixregister ist.

16. Verschachteler (2) nach Anspruch 14 oder 15, dadurch gekennzeichnet, daß aus den beiden Schieberegistern jeweils zwei Bit parallel in das Matrixregister geschrieben werden.

17. Verschachteler (2) nach Anspruch 14 oder 15, dadurch gekennzeichnet, daß die beiden Schieberegister, nachdem sie durch Eingaben über die entsprechenden Dateneingänge des Verschachtelers (2) vollständig gefüllt wurden, ihre Bits gemeinsam in kammartig verzahnter Weise als Bitspalte in das Matrixregister eingeben und auf diese Weise nach und nach mehrere oder alle Spalten des Matrixregisters füllen.

18. Verschachteler (2) nach Anspruch 10, dadurch gekennzeichnet, daß der Verschachteler (2) einen RAM aufweist und so eingerichtet ist, daß die in den Verschachteler (2) gelangenden Bitpaare direkt auf vorbestimmte RAM-Adressen geschrieben werden.

19. Elektronische Sendervorrichtung mit einer Punktierereinrichtung nach Anspruch 1, wobei die Sendervorrichtung einen in Datenstromrichtung hinter der Punktierereinrichtung angeordneten Verschachteler (2) aufweist, der

- einen ersten Dateneingang, welcher mit dem ersten Datenausgang der Punktierereinrichtung direkt oder indirekt elektrisch verbunden ist, und
- einen zweiten Dateneingang, welcher mit dem zweiten Datenausgang der Punktierereinrichtung direkt oder indirekt elektrisch verbunden ist, aufweist.

20. Elektronische Sendervorrichtung nach Anspruch 19, dadurch gekennzeichnet, daß

- der Verschachteler (2) so ausgelegt ist, daß er mit Hilfe des genannten, von der Punktierereinrichtung zusätzlich übertragenen Signals (data_valid) die leeren Plätze im von der Punktierereinrichtung kommenden parallelen Eingabedatenstrom erkennt und bei der weiteren Datenverarbeitung nicht beachtet.

21. Entschachteler (3) zur Verwendung in einer elektronischen Empfängervorrichtung, wobei der Entschachteler (3)

- einen ersten und einen zweiten Datenausgang aufweist,
- so ausgelegt ist, daß er seinen Ausgabedatenstrom im Wesentlichen gleichmäßig parallel auf seine beiden genannten Datenausgänge verteilt,
- ein Blockdeinterleaver ist und
- einen einzigen zusammenhängenden Speicherblock mit Zeilen und Spalten umfasst.

22. Entschachteler (3) nach Anspruch 21, dadurch gekennzeichnet, daß der Entschachteler (3) ein $n \times m$ -Entschachteler ist, wobei n und m natürliche Zahlen sind.

23. Entschachteler (3) nach Anspruch 22, dadurch gekennzeichnet, daß der Entschachteler (3) ein Matrixregister aufweist.

24. Entschachteler (3) nach Anspruch 23, dadurch gekennzeichnet, daß das Matrixregister ein $16 \times (18 \times N)$ -Matrixregister ist, wobei N die Wortlänge der Softbits ist.

25. Entschachteler (3) nach Anspruch 23 oder 24, dadurch gekennzeichnet, daß der Entschachteler (3) so ausgelegt ist, daß jeweils zwei Softbits parallel aus dem Matrixregister ausgelesen werden.

26. Entschachteler (3) nach einem der Ansprüche 22 bis 25, dadurch gekennzeichnet, daß der Entschachteler (3) ein elektrisch direkt oder indirekt an seinen ersten Datenausgang angeschlossenes erstes Schieberegister und ein elektrisch direkt oder indirekt an seinen zweiten Datenausgang angeschlossenes zweites Schieberegister aufweist, wobei die beiden genannten Schieberegister als Softbit-Schieberegister ausgelegt sind.

27. Entschachteler (3) nach auf Anspruch 22 rückbezogenem Anspruch 26, dadurch gekennzeichnet, daß beide Schieberegister $\frac{1}{2}n$ -Softbit-Schieberegister sind.

28. Entschachteler (3) nach Anspruch 26 oder 27, dadurch gekennzeichnet, daß der Entschachteler (3) so ausgelegt ist, daß beim Ausgeben der Daten aus der $n \times m$ -Struktur oder aus dem Matrixregister zunächst eine Spalte in kammartig verzahnter Weise, d. h. derart an die beiden Schieberegister ausgegeben wird, daß zwei benachbarte Softbits jeweils an ein anderes Schieberegister gelangen, danach beide Schieberegister gleichzeitig ausgelesen und nach dem Auslesen der beiden Schieberegister nacheinander weitere Datenspalten aus der $n \times m$ -Struktur oder aus dem Matrixregister in gleicher Art wie bei der ersten ausgegebenen Spalte an die beiden Schieberegister ausgegeben werden.

29. Entschachteler (3) nach Anspruch 21, dadurch gekennzeichnet, daß der Entschachteler (3) einen RAM

aufweist und so ausgelegt ist, daß bei der Datenausgabe die Bitpaare aus dem RAM direkt an die beiden Datenausgänge des Entschachtlers (3) gelangen.

30. Elektronische Empfängervorrichtung, mit

- einem Entschachtler (3) und
- einer in Datenstromrichtung hinter dem Entschachtler (3) angeordneten Depunktierereinrichtung, wobei der Entschachtler (3),
- einen ersten und einen zweiten Datenausgang aufweist,
- so ausgelegt ist, daß er seinen Ausgabedatenstrom im Wesentlichen gleichmäßig parallel auf seine beiden genannten Datenausgänge verteilt,
- leere Plätze in seinem Ausgabedatenstrom vorsieht, so dass die Anzahl der Bits des Ausgangsdatenstroms des Entschachtlers (3) einschließlich der leeren Plätze der Anzahl der Bits des Ausgangsdatenstroms der Depunktierereinrichtung entspricht, und
- zusätzlich zu seinem parallelen Ausgabedatenstrom der Depunktierereinrichtung ein Signal (data_valid) übermittelt, welches die Depunktierereinrichtung über leere Plätze im parallelen Ausgabedatenstrom des Entschachtlers (3) informiert.

31. Elektronische Empfängervorrichtung nach Anspruch 30, dadurch gekennzeichnet, dass die Depunktierereinrichtung zwei Dateneingänge aufweist, wobei der erste Dateneingang der Depunktierereinrichtung mit dem ersten Datenausgang des Entschachtlers (3) elektrisch direkt oder indirekt verbunden ist und der zweite Dateneingang der Depunktierereinrichtung mit dem zweiten Datenausgang des Entschachtlers (3) elektrisch direkt oder indirekt verbunden ist.

32. Elektronische Empfängervorrichtung nach Anspruch 31, dadurch gekennzeichnet, daß

- die Depunktierereinrichtung so ausgelegt ist, daß sie mit Hilfe des genannten, vom Entschachtler (3) zusätzlich übertragenen Signals (data_valid) die leeren Plätze im vom Entschachtler (3) kommenden parallelen Eingabedatenstrom erkennt und bei der weiteren Datenverarbeitung mit Soft-Nullen auffüllt.

33. Depunktierereinrichtung zur Verwendung in einer elektronischen Empfängervorrichtung, wobei die Depunktierereinrichtung zwei Dateneingänge aufweist und so ausgelegt ist, daß sie auf beiden Dateneingängen parallel einlaufende Datenströme verarbeiten kann und einen ersten Depunktierer (P2') und einen in Datenstromrichtung hinter dem ersten Depunktierer (P2') angeordneten zweiten Depunktierer (P1') aufweist, wobei

- der erste Depunktierer (P2') leere Plätze in seinem Ausgabedatenstrom vorsieht, so dass die Anzahl der Bits des Ausgangsdatenstroms des ersten Depunktierers (P2') einschließlich der leeren Plätze der Anzahl der Bits des Ausgangsdatenstroms des zweiten Depunktierers (P1') entspricht, und
- der erste Depunktierer (P2') so ausgelegt ist, daß er zusätzlich zu seinem parallelen Ausgabedatenstrom dem zweiten Depunktierer (P1') ein Signal (data_valid) übermittelt, welches den zweiten Depunktierer (P1') über leere Plätze im parallelen Ausgabedatenstrom des ersten Depunktierers (P2') informiert.

34. Depunktierereinrichtung nach Anspruch 33, dadurch gekennzeichnet, daß der erste Depunktierer (P2')

- einen ersten Multiplexer (MUX) mit zwei Eingängen und einem Ausgang,
- einen zweiten Multiplexer (MUX) mit zwei Eingängen und einem Ausgang und
- einen dritten Multiplexer (MUX) mit vier Eingängen und einem Ausgang aufweist, zwischen
- den Ausgang des ersten Multiplexers (MUX) und einen Eingang des zweiten Multiplexers (MUX), den Ausgang des zweiten Multiplexers (MUX) und einen ersten Datenausgang (Out_X) des ersten Depunktierers (P2'),
- den Ausgang des dritten Multiplexers (MUX) und einen zweiten Datenausgang (Out_Y) des ersten Depunktierers (P2') und
- einen ersten Dateneingang (IN_Y) des ersten Depunktierers (P2') und einen Eingang des dritten Multiplexers (MUX) jeweils ein 1-Schritt-Verzögerungsregister (D) geschaltet ist und
- der erste Dateneingang (IN_Y) des ersten Depunktierers (P2') elektrisch unmittelbar ferner an einen Eingang des ersten Multiplexers (MUX) und an einen weiteren Eingang des dritten Multiplexers (MUX) angeschlossen ist,
- der zweite Dateneingang (IN_X) des ersten Depunktierers (P2') elektrisch unmittelbar an den weiteren Eingang des zweiten Multiplexers (MUX) und den dritten Eingang des dritten Multiplexers (MUX) angeschlossen ist und
- der jeweils verbleibende Eingang des ersten Multiplexers (MUX) und des dritten Multiplexers (MUX) an eine Leitung angeschlossen ist, auf der Soft-Nullen bereitgestellt werden.

35. Depunktierereinrichtung nach Anspruch 33 oder Anspruch 34, dadurch gekennzeichnet, daß der zweite Depunktierer (P1') drei Multiplexer (MUX) mit jeweils zwei Eingängen und einem Ausgang aufweist, zwischen

- den Ausgang des ersten Multiplexers (MUX) und einen Eingang des zweiten Multiplexers (MUX),
- den Ausgang des zweiten Multiplexers (MUX) und den ersten Datenausgang (Out_X) des zweiten Depunktierers (P1') und
- den Ausgang des dritten Multiplexers (MUX) und den zweiten Datenausgang (Out_Y) des zweiten Depunktierers (P1')

jeweils ein 1-Schritt-Verzögerungsregister (D) geschaltet ist und

- der erste Dateneingang (IN_X) des zweiten Depunktierers (P1') elektrisch unmittelbar an einen Eingang des ersten Multiplexers (MUX) und an den weiteren Eingang des zweiten Multiplexers (MUX) angeschlossen ist,
- der zweite Dateneingang (IN_Y) des zweiten Depunktierers (P1') elektrisch unmittelbar an einen Eingang des dritten Multiplexers (MUX) angeschlossen ist und
- der jeweils verbleibende Eingang des ersten und des dritten Multiplexers (MUX) an eine Leitung angeschlossen ist, auf der Soft-Nullen bereitgestellt werden.

36. Depunktierereinrichtung nach einem der Ansprüche 33 bis 35, dadurch gekennzeichnet, daß

- der erste Depunktierer (P2') einen ersten und einen zweiten Datenausgang aufweist und so ausgelegt ist, daß er seinen Ausgabedatenstrom im Wesentlichen gleichmäßig auf seine beiden Datenausgänge verteilt, und
- der zweite Depunktierer (P1') einen ersten und einen zweiten Dateneingang aufweist, wobei der erste Dateneingang des zweiten Depunktierers (P1') elektrisch direkt oder indirekt mit dem ersten Datenausgang des ersten Depunktierers (P2') und der zweite Dateneingang des zweiten Depunktierers (P1') elektrisch direkt oder indirekt mit dem ersten Datenausgang des ersten Depunktierers (P2') verbunden ist.

37. Depunktierereinrichtung nach einem der Ansprüche 33 bis 36, dadurch gekennzeichnet, daß

- der zweite Depunktierer (P1') so ausgelegt ist, daß er mit Hilfe des genannten, vom ersten Depunktierer (P2') zusätzlich übertragenen Signals (data_valid) die leeren Plätze im vom ersten Depunktierer (P2') kommenden parallelen Eingabedatenstrom erkennt und bei der weiteren Datenverarbeitung mit Soft-Nullen auffüllt.

38. Depunktierereinrichtung nach einem der Ansprüche 33 bis 37, dadurch gekennzeichnet, daß der erste

Depunktierer (P2') zwei Dateneingänge aufweist.

39. Depunktierereinrichtung nach Anspruch 38, dadurch gekennzeichnet, daß die beiden Dateneingänge

des ersten Depunktierers (P2') gleichzeitig die beiden Dateneingänge der Depunktierereinrichtung sind.

40. Nachrichtenübertragungssystem, dadurch gekennzeichnet, daß es eine Punktiervorrichtung, einen Verschachteler (2) oder eine elektronische Sendervorrichtung nach einem der Ansprüche 1 bis 9, 10 bis 18 oder 19 bis 20 und/oder einen Entschachteler (3), eine elektronische Empfängervorrichtung oder eine Depunktierereinrichtung nach einem der Ansprüche 21 bis 29, 30 bis 32 oder 33 bis 39 aufweist.

41. Nachrichtenübertragungssystem nach Anspruch 40, dadurch gekennzeichnet, daß es so ausgelegt ist, daß die Übertragung zwischen Sender und Empfänger drahtlos vonstatten geht.

42. Nachrichtenübertragungssystem nach Anspruch 41, dadurch gekennzeichnet, daß es ein WLAN ist.

43. Nachrichtenübertragungssystem nach einem der Ansprüche 40 bis 42, dadurch gekennzeichnet, daß die Taktfrequenz des Systems im Bereich von 75 MHz bis 85 MHz liegt.

44. Nachrichtenübertragungssystem nach Anspruch 43, dadurch gekennzeichnet, daß die Taktfrequenz des Systems 80 MHz beträgt.

Es folgen 15 Blatt Zeichnungen

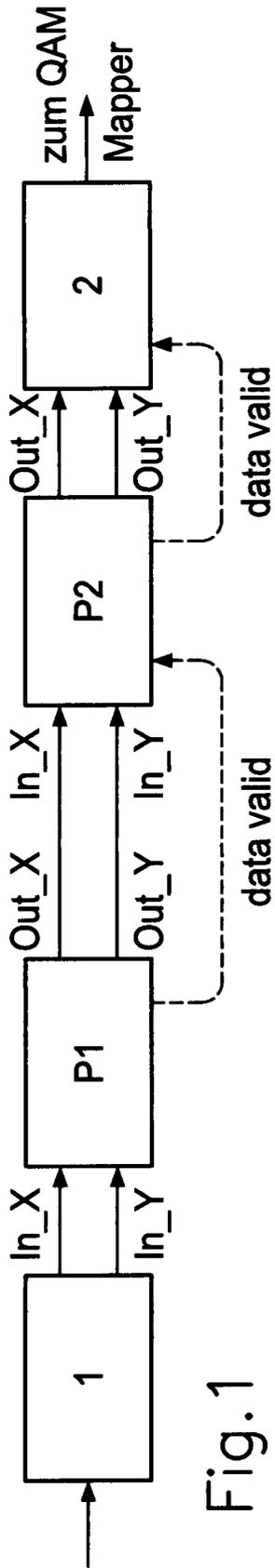


Fig. 1

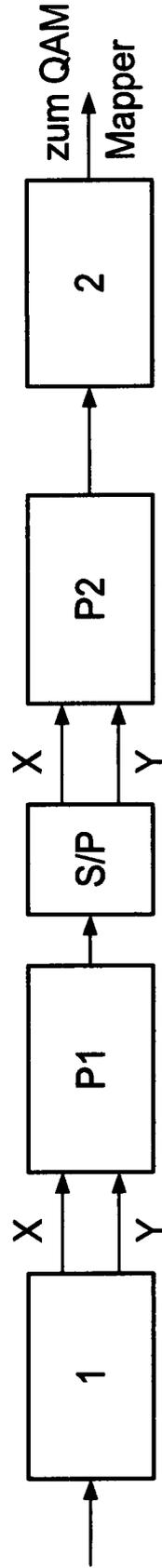


Fig. 2

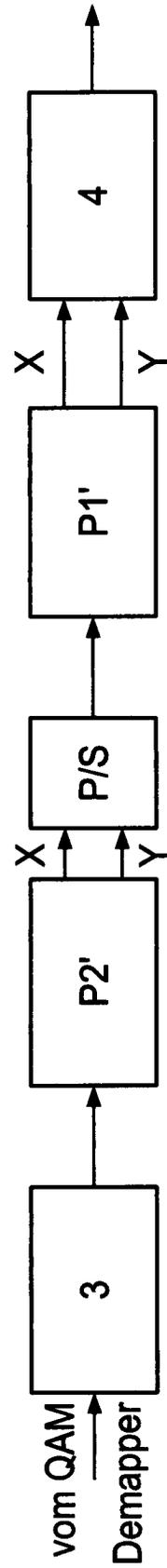


Fig. 3

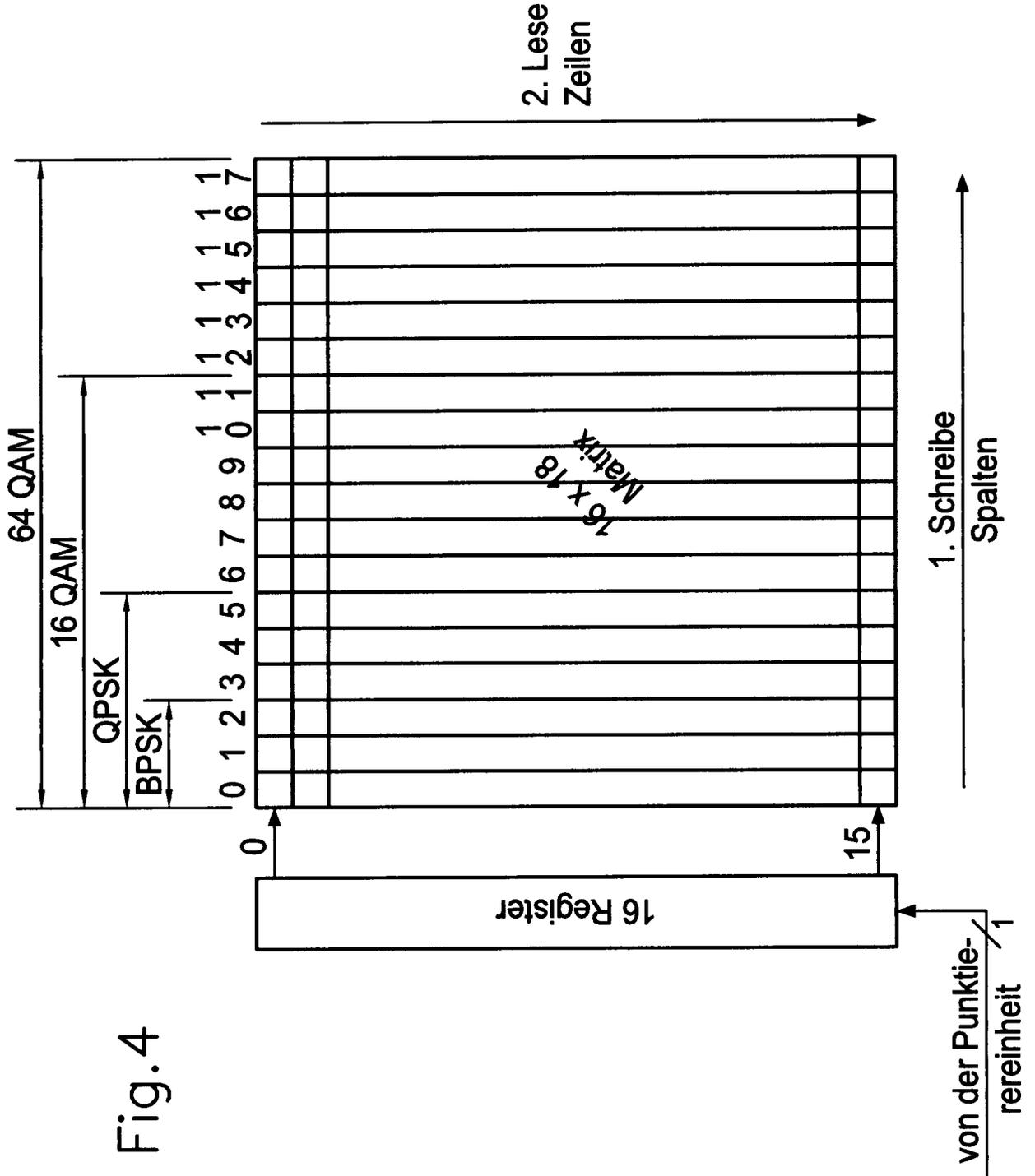


Fig. 4

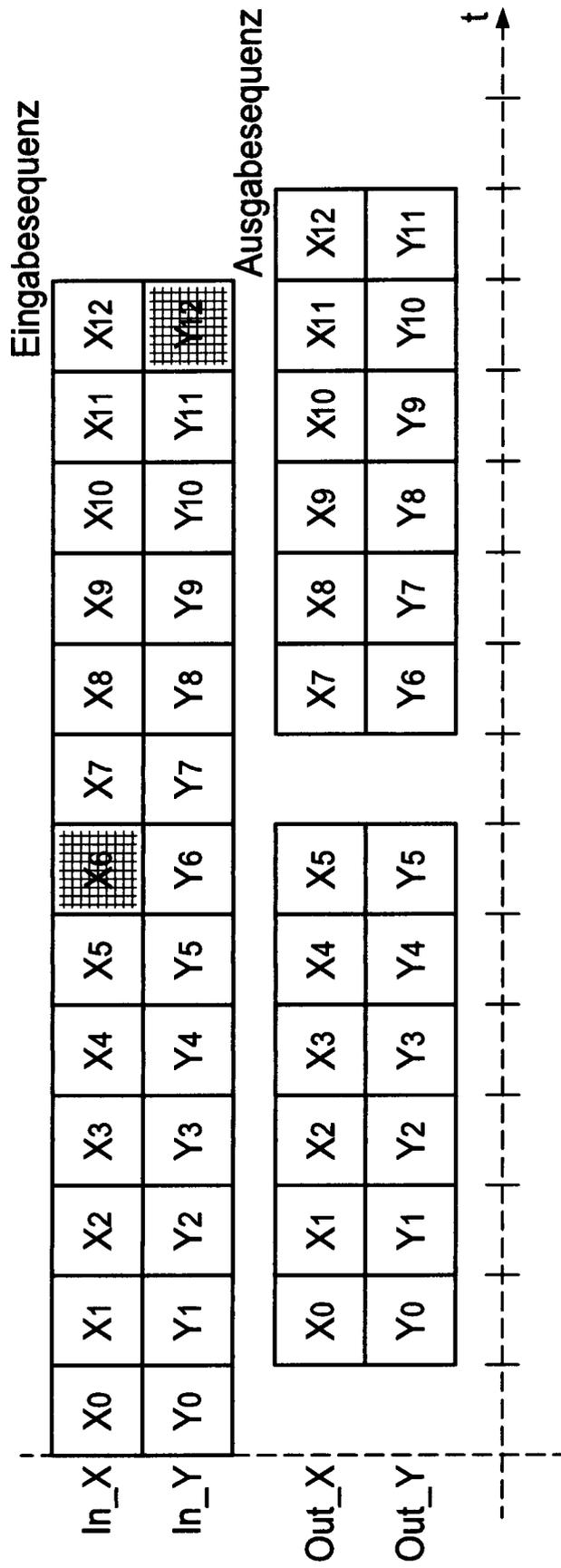


Fig. 5

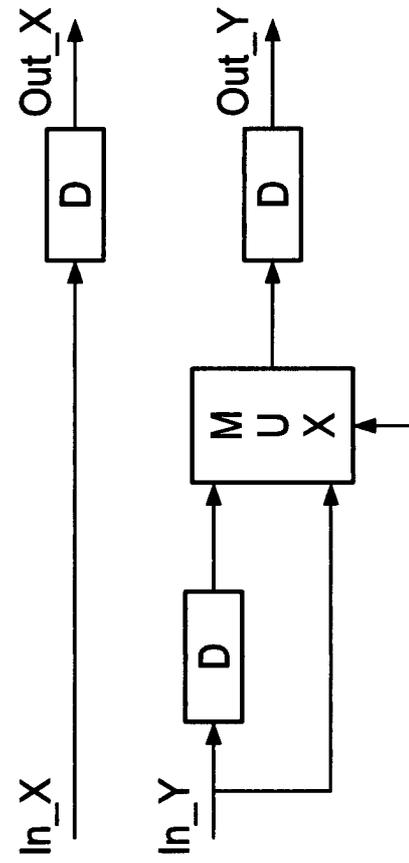


Fig. 6

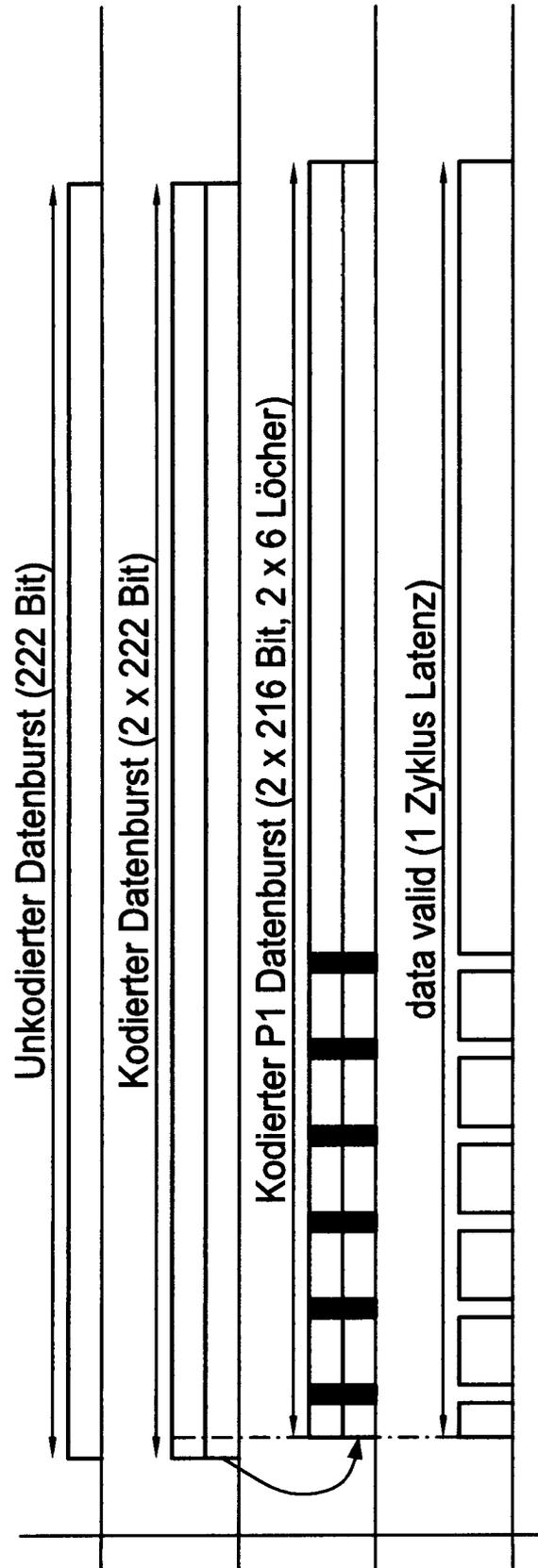


Fig.7

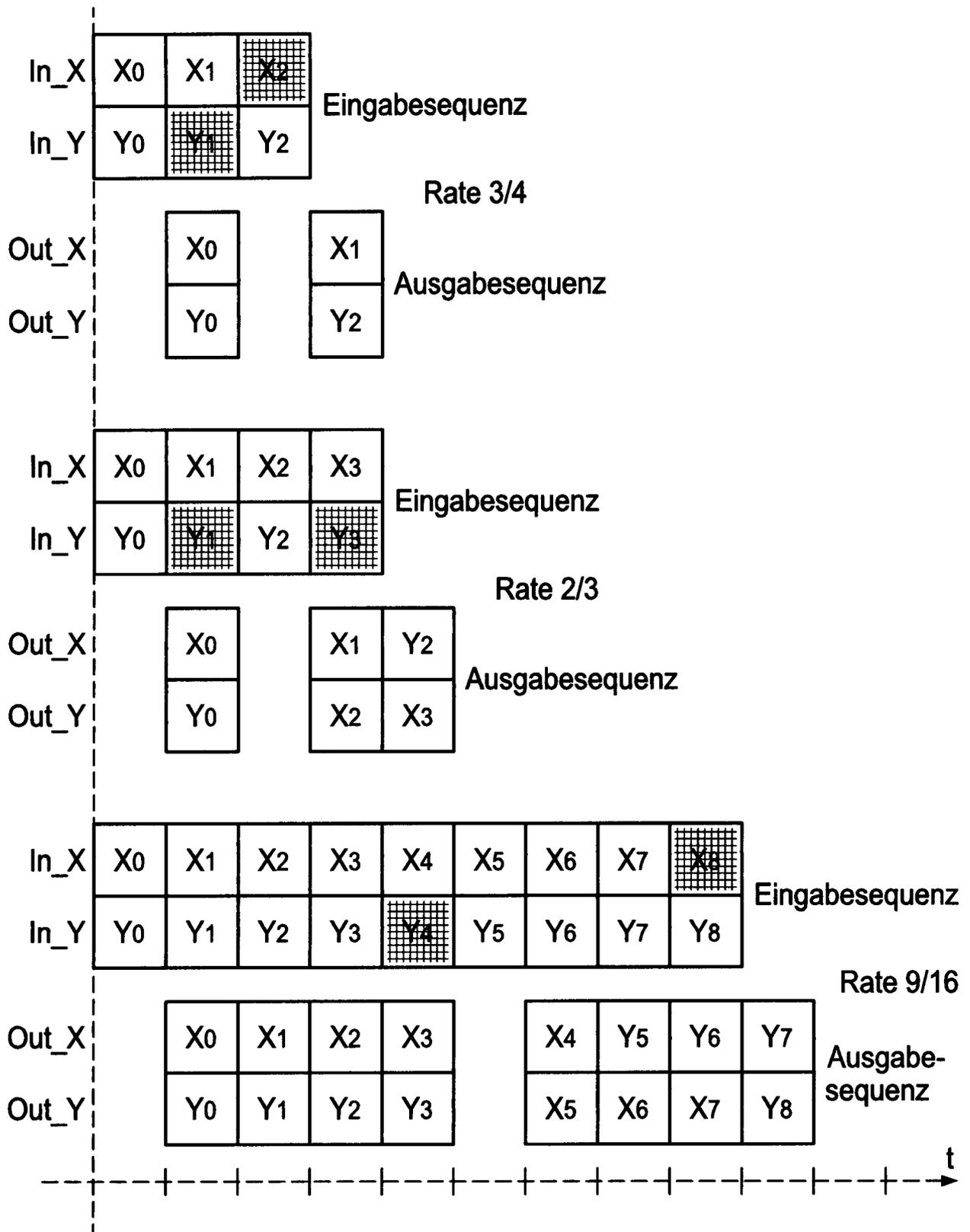


Fig.8

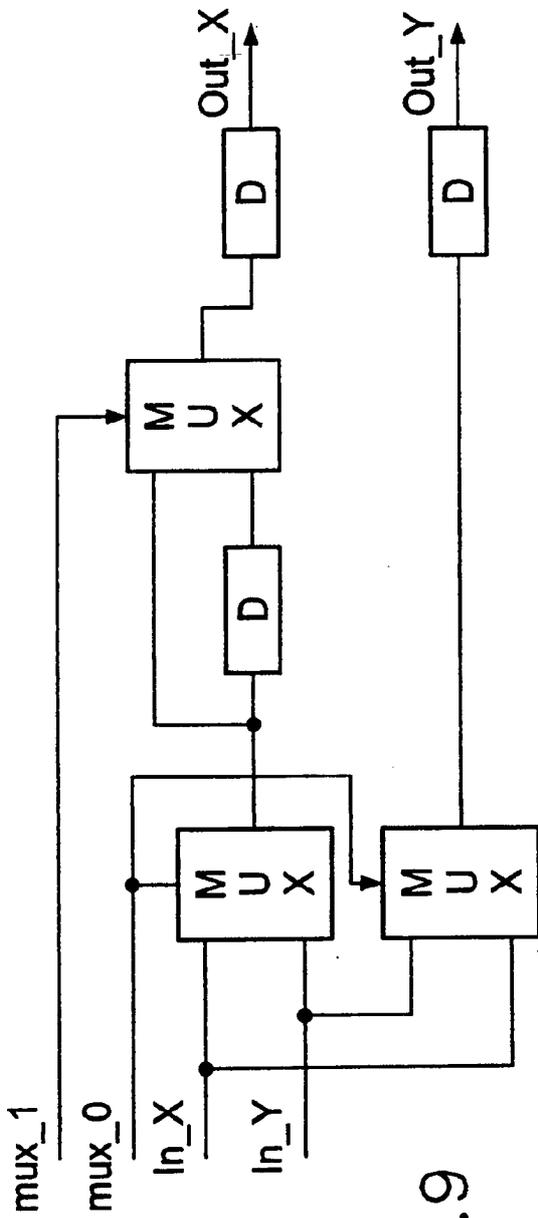


Fig.9

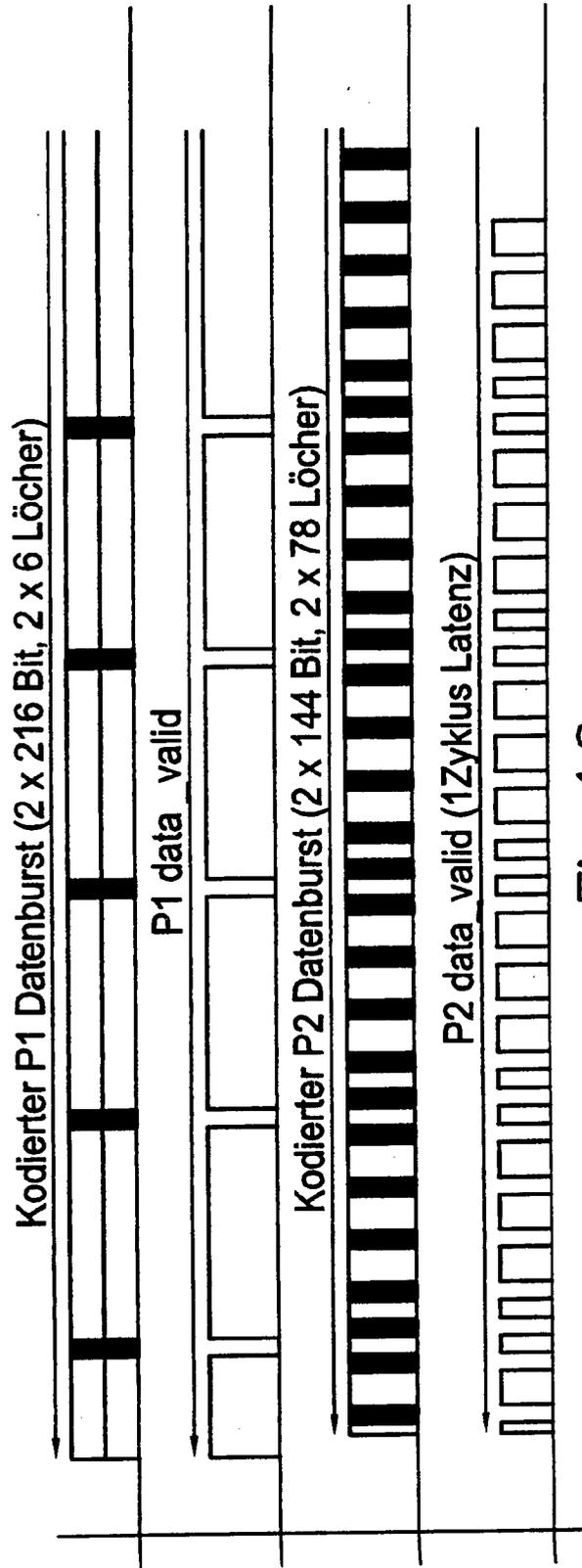


Fig.10

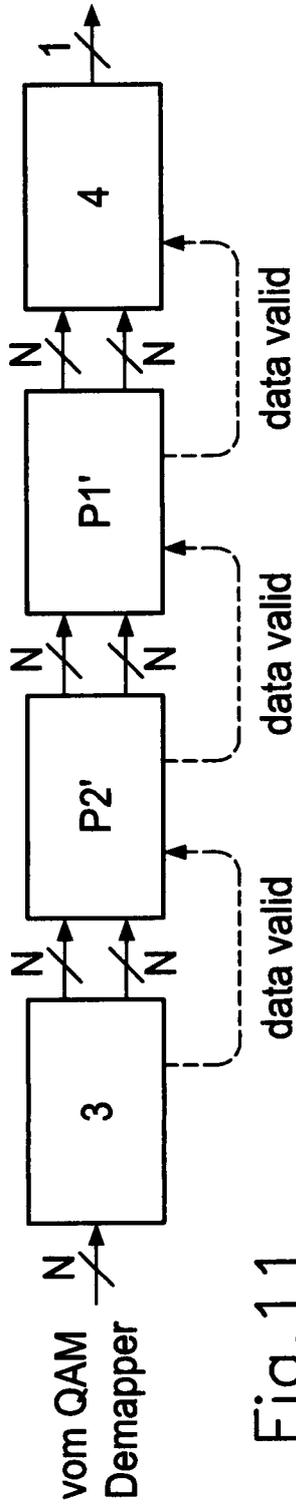


Fig.11

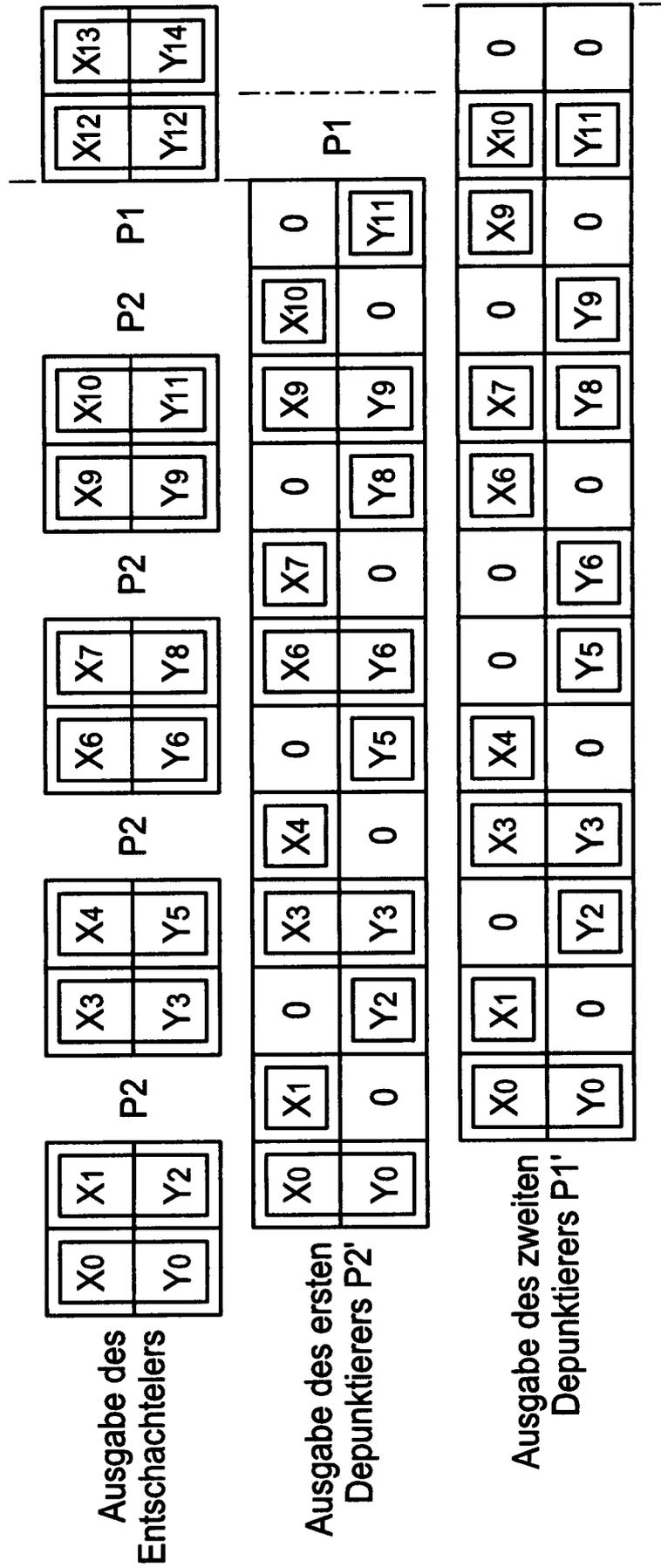


Fig.12

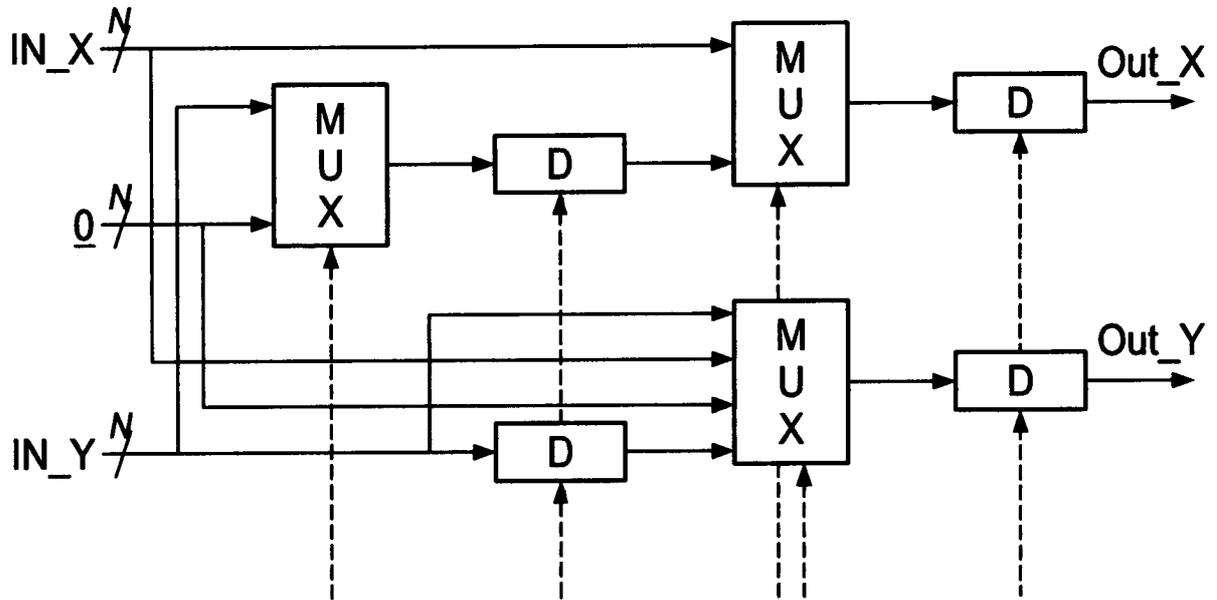


Fig.13(a)

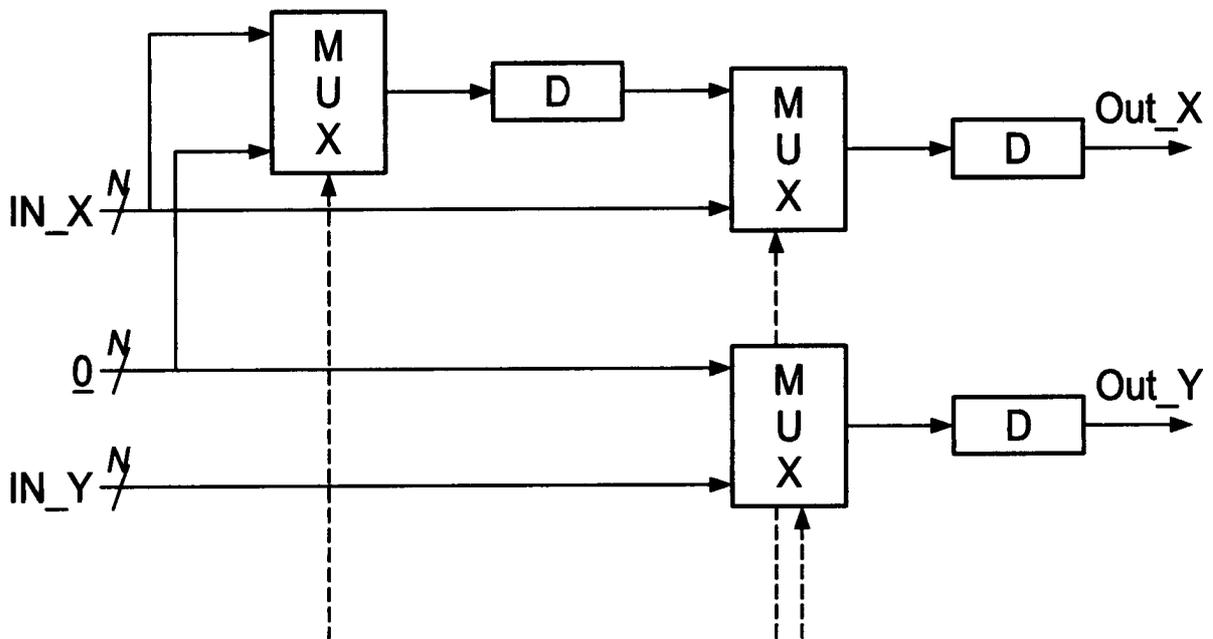


Fig.13(b)

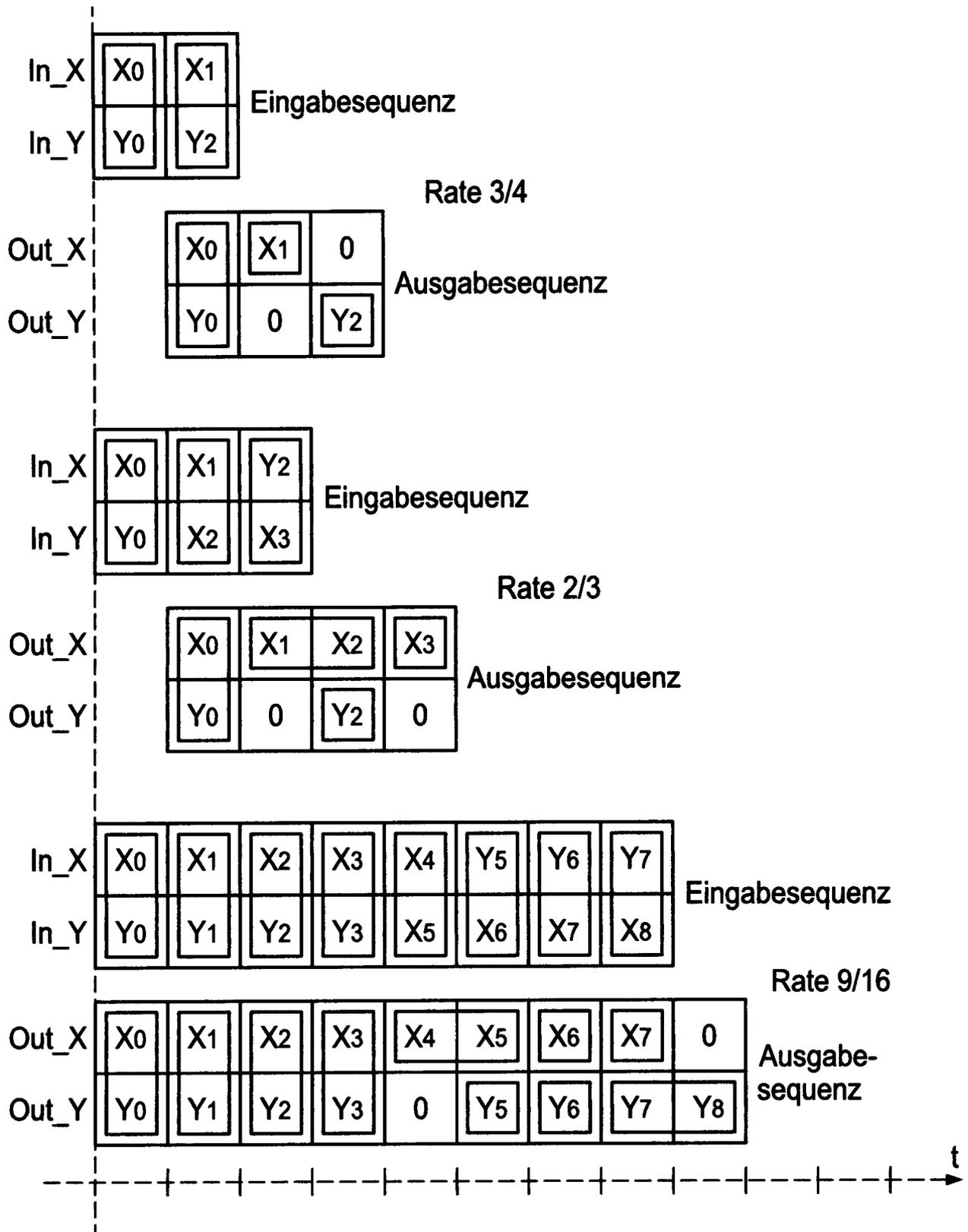
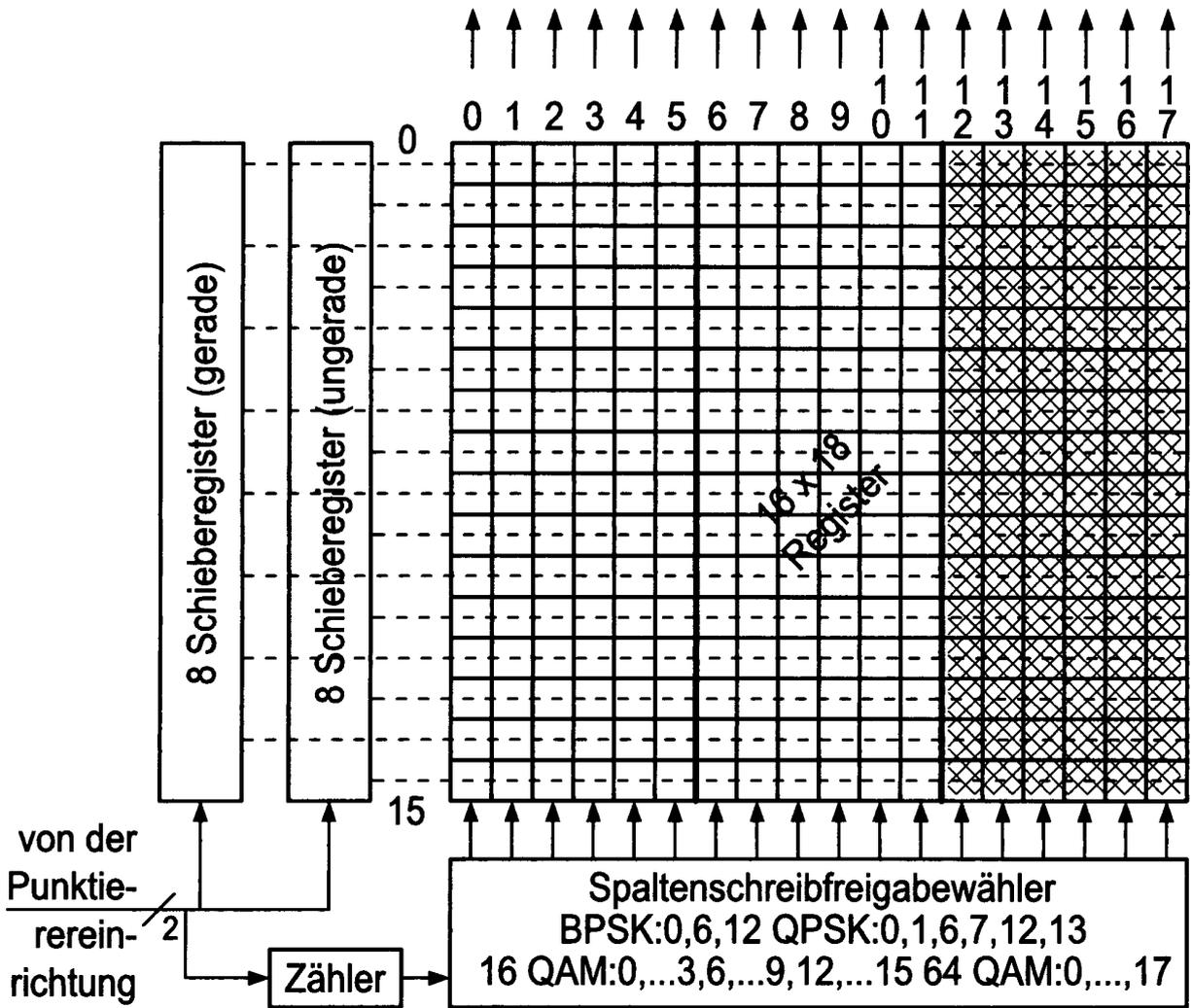
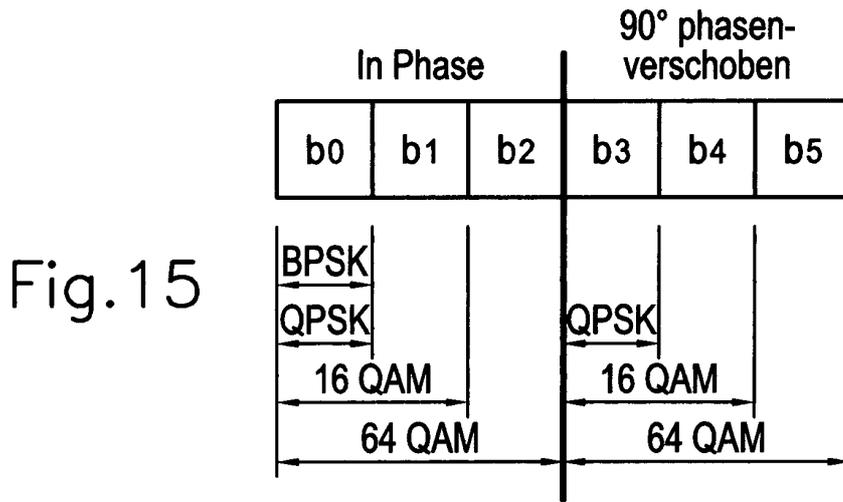


Fig.14



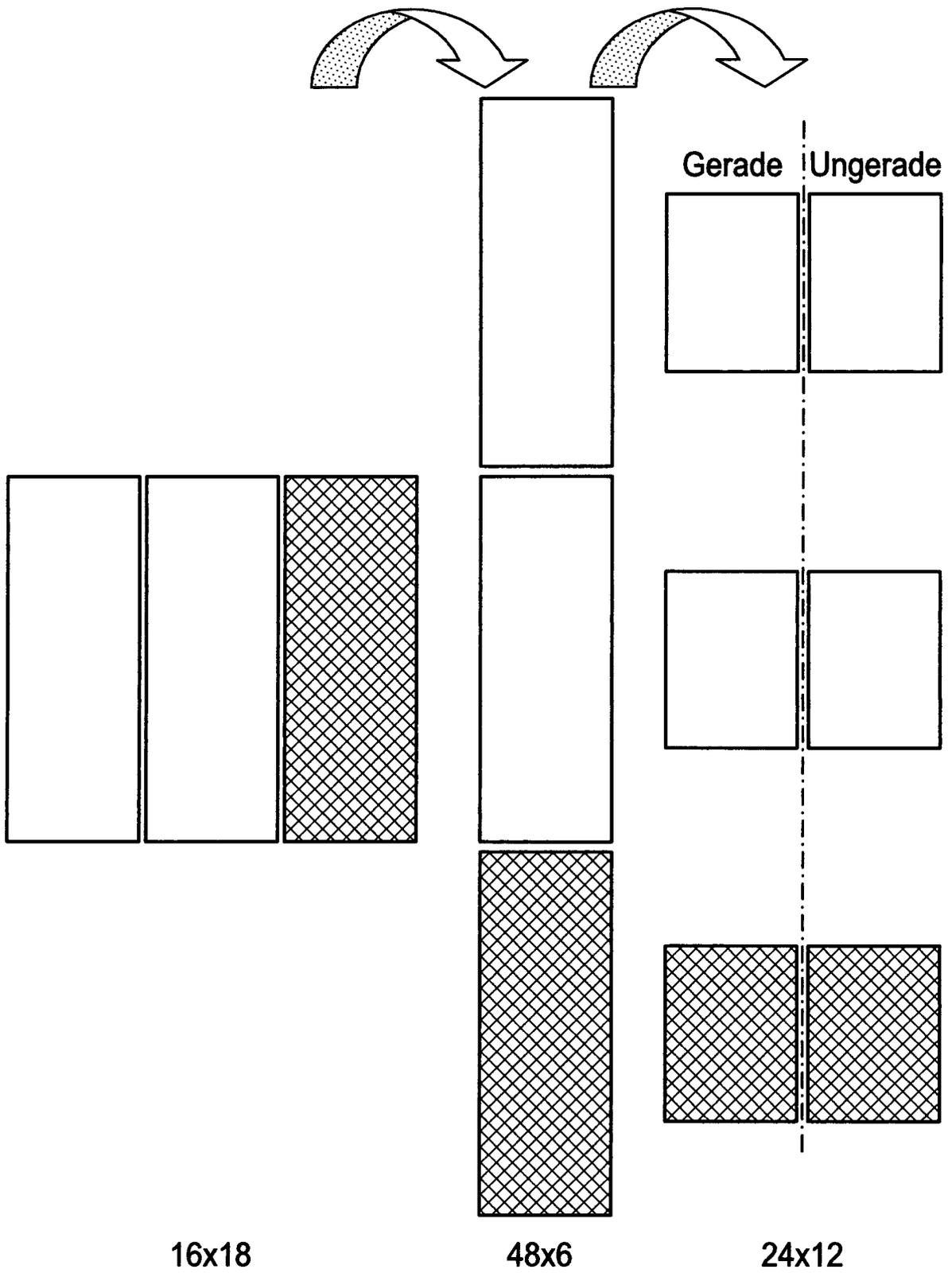


Fig.17

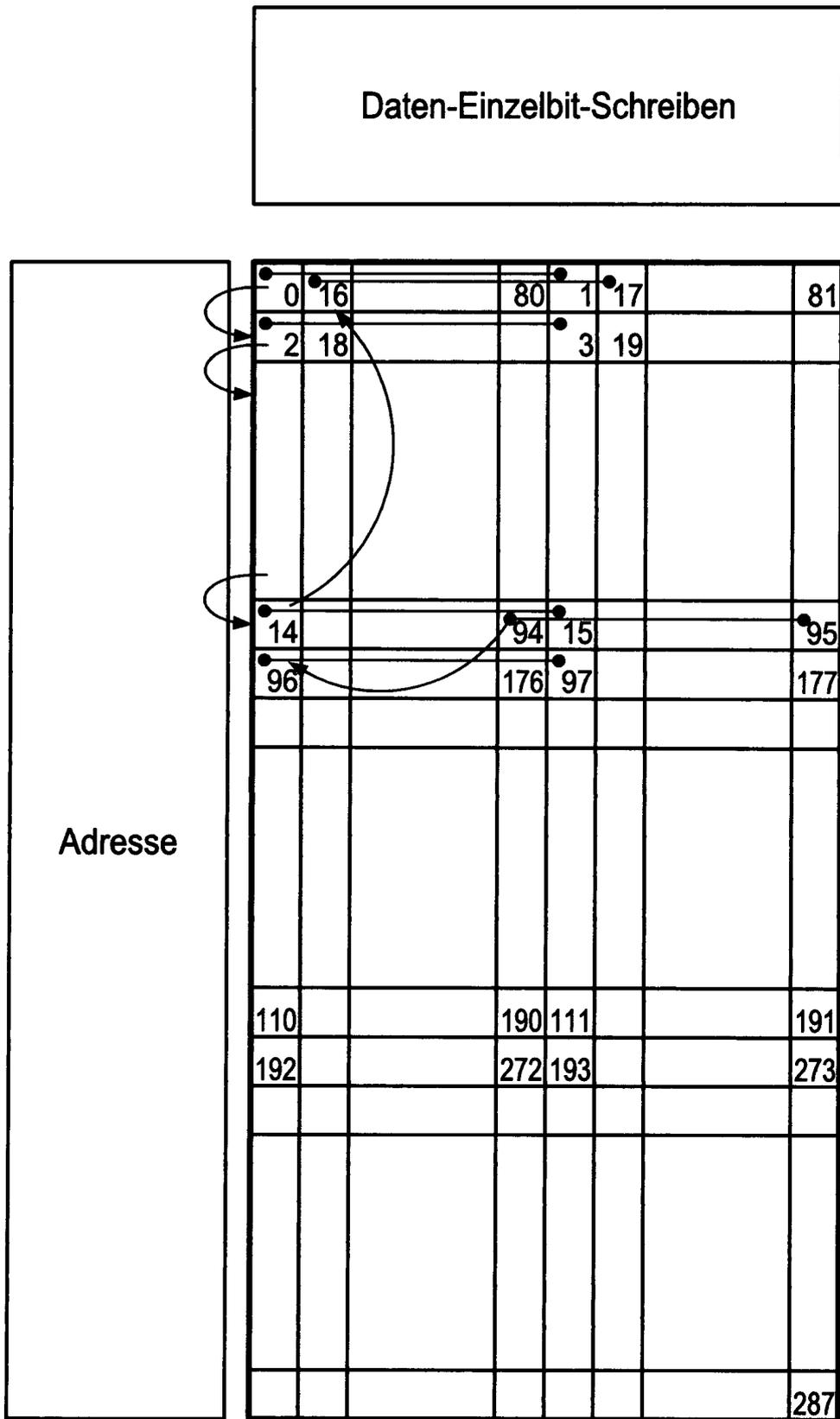


Fig.18

**12 Bit Lesen +
MUX (6MSB-6LSB)**

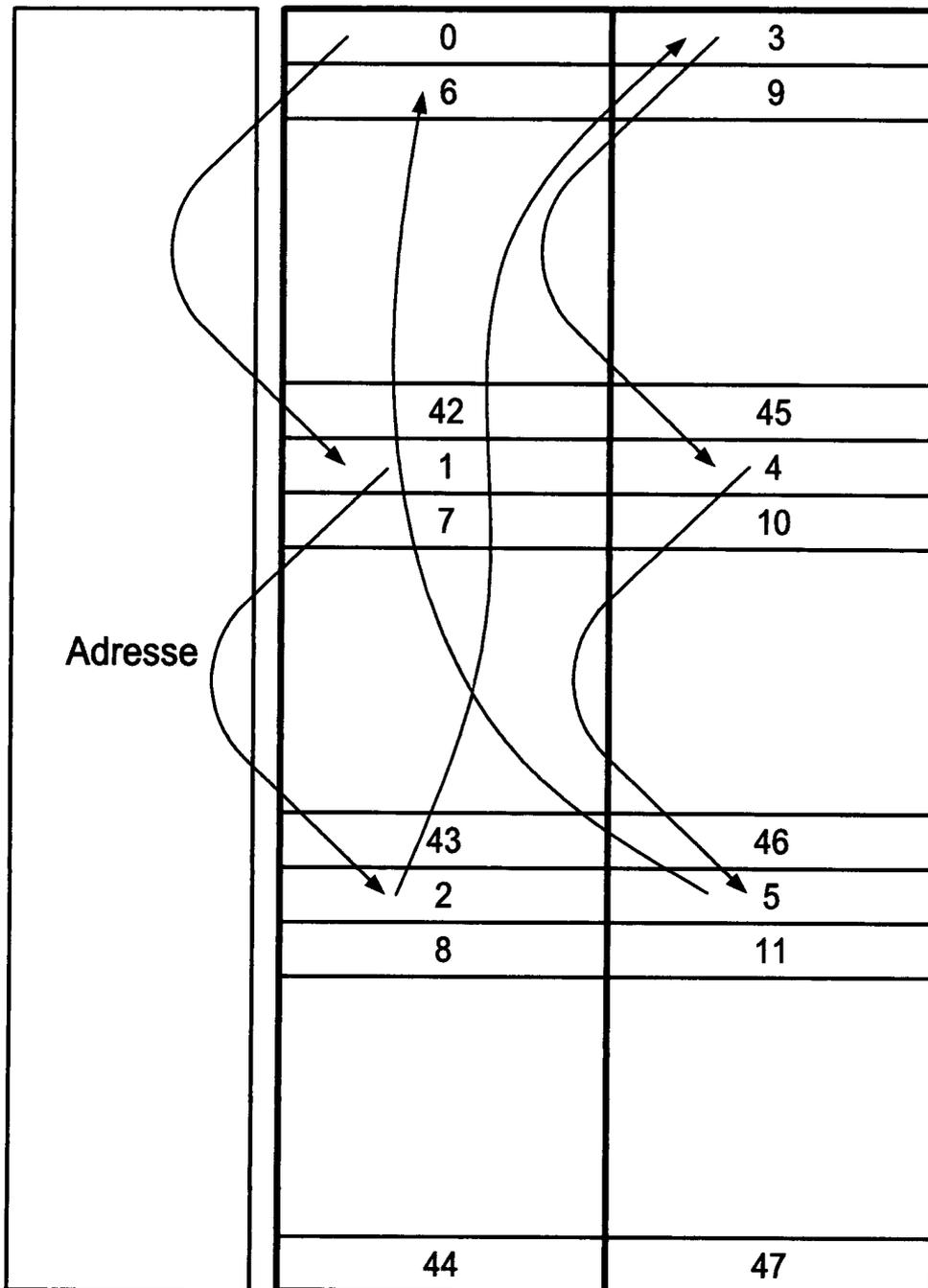


Fig.19

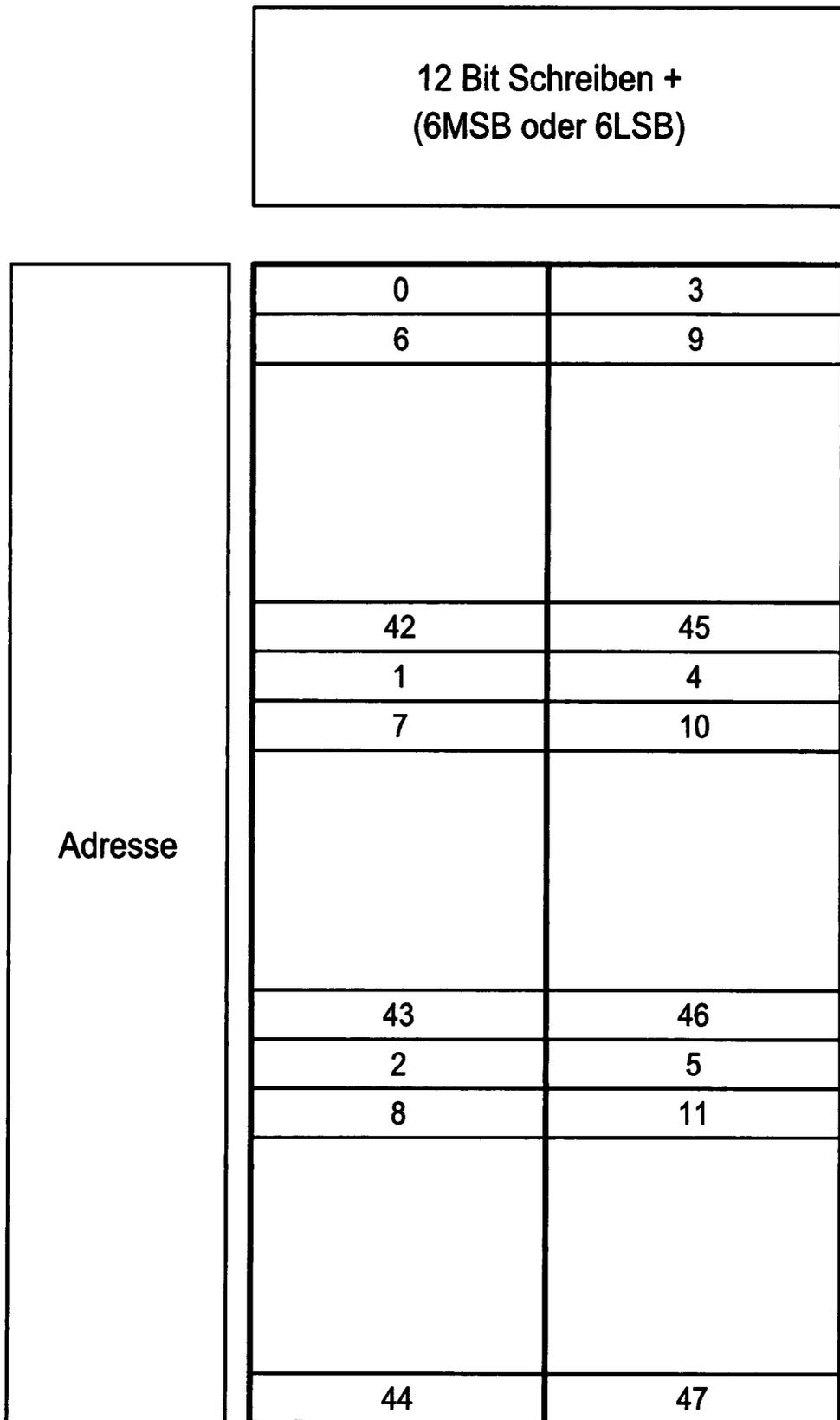


Fig.20

Lesen + Paarauswahl

Adresse	0	16		80	1	17		241
	2	18			3	19		
	14			94	15			95
	96			176	97			177
	110			190	111			191
	192			272	193			272
								287

Fig.21