Theoretical Analysis of M-ary/SS Communication Systems Using Racing Counters and a Hadamard Matrix

Kouji Ohuchi, Hiromasa Habuchi, and Takaaki Hasegawa, *Member, IEEE*

***Abstract*— In this paper, performance of M-ary spread spectrum (M-ary/SS) communication systems is discussed. Firstly, the initial acquisition time is evaluated. Secondly, retention time, which is the average number of frames holding correct frame timing, and recovery time, which is the average number of frames required to establish synchronization, are derived. Lastly, bit-error rate (BER) performance is evaluated. M-ary/SS communication systems, which have more than one spreading code, can improve BER performance under conditions in which there is additive white Gaussian noise (AWGN). However, the synchronization of M-ary/SS communication systems is difficult because they have several spreading codes. The frame synchronization method which is used in this paper uses a Hadamard matrix and "racing counters." As a result, the retention time becomes longer than the recovery time when the size of the lower counter differs greatly from that of the upper counter in the racing counters. Then BER gets close to the performance which is achieved under the complete synchronization.**

I. Introduction

Spread spectrum (SS) communication systems are resistant to interference and have a low probability of interception. M-ary spread spectrum (M-ary/SS) communication systems, which have more than one spreading code, can improve bit-error rate (BER) performance under conditions in which there is additive white Gaussian noise (AWGN) [1]–[6]. However, the synchronization of M-ary/SS communication systems is difficult because they have several spreading codes. There is no investigation of performance of the M-ary/SS communication system by taking account of synchronization performance. Much effort has therefore gone into developing appropriate synchronization methods. There is, for example, a system using several delay lock loops [4], [5] and a scheme using a synchronizing spreading code [6]. Moreover, we have proposed a system using a Hadamard matrix and the racing counters [7], [8].

In this paper, we discuss performance of M-ary/SS communication systems by considering the frame synchronization. The frame synchronization system which is used in this paper uses a Hadamard matrix and the racing counters [7], [8]. It has a very simple structure, and it achieves initial synchronization without a preamble code for acquisition. Racing counters have been used on the frame synchronization of conventional pulse code modulation (PCM) communication systems [9], [10]. The frame synchronization system uses the row vectors of a Hadamard matrix as the spreading code. The head chip of each code is always plus. Since the head chip of each code is used for the frame synchronization, it is possible to implement a simple synchronization system without extra synchronizing signals. In this paper, we evaluate performance of the M-ary/SS communication system using a Hadamard matrix and the racing counters by a theoretical analysis and computer simulations. In particular, we discuss the BER performance that took account of the synchronization performance.

In Section II, we describe the structure of the frame synchronization system [7], [8]. In Section III, we discuss the performance of the frame synchronization system. Initial synchronization method is divided into the following two methods: 1) the method using preambles and 2) the method using no preambles. Firstly, we evaluate initial acquisition time of the system with preambles and the system without preambles. Secondly, we derive retention time, which is the average number of frames holding correct frame timing, and recovery time, which is the average number of frames required to establish synchronization. In Section IV, we analyze BER performance. Finally, we summarize the main results and the future problems in Section V.

Table I shows the notation used in the following discussion.

II. A Synchronization System

A. Hadamard Matrix

A Hadamard matrix is defined as

$$H_K = \begin{bmatrix} H_{K-1} & H_{K-1} \\ H_{K-1} & \overline{H_{K-1}} \end{bmatrix}. \tag{1}$$

For example, when $K = 2$, we have following matrix:

$$H_2 = \begin{bmatrix} + & + & + & + \\ + & - & + & - \\ + & + & - & - \\ + & - & - & + \end{bmatrix} = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{bmatrix}. \tag{2}$$

(head chips ↓)

The row vectors $a_i (i = 1, \cdots, 4)$ of the Hadamard matrix are used as the spreading code. The correlation value between any

Manuscript received May 1, 1995; revised October 3, 1995.

K. Ohuchi and H. Habuchi are with the Department of Computer and Information Sciences, Faculty of Engineering, Ibaraki University, Hitachi, Ibaraki, Japan.

T. Hasegawa is with the Department of Electrical and Electronic Engineering, Faculty of Engineering, Saitama University, Urawa, Saitama, Japan.

Publisher Item Identifier S 0733-8716(96)05207-9.

TABLE I
NOTATIONS

Number of stages of counter C2	m
Number of stages of counter C3	n
Probability of holding frame timing	P_h
Probability of renewing frame timing	P_r
Probability of renewing incorrect frame timing	P_R
Average number of frames required until frame timing is renewed	T_h
Average number of frames required to return to correct frame timing (without preambles)	T_{r1}
Average number of frames required to return to correct frame timing (with preambles)	T_{r2}
Initial acquisition time [frame] (without preambles)	N_{acq1}
Initial acquisition time [frame] (with preambles)	N_{acq2}
Retention time [frame]	N_h
Recovery time [frame]	N_r

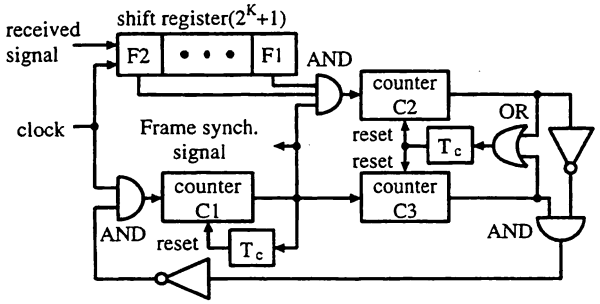

Fig. 1. Frame synchronization system.

two codes is zero. We take notice that the head chip of each code is always plus. The head chip of each code is therefore used as a framing pulse. This makes it possible to use of the frame synchronization method of a PCM communication system without extra codes for synchronization.

B. Frame Synchronization Method

The frame synchronization system uses "racing counters" which have been used in conventional PCM communication systems [9]. The frame synchronization system consists of one shift register and three counters (C1, C2, and C3) as shown in Fig. 1, where "T_c" means chip duration.

The function of each component in the frame synchronization system is as follows.

- *The shift register* stores the received signal for one frame plus one chip ($=2^K+1$ chips, 2^K is the length of the spreading code). When the frame timing is correct, it can detect the head chip of each frame at the position of F1 and F2.
- C1 counts until it reaches a value of 2^K. The output signal from C1 can be used as a frame synchronizing signal.
- C2 counts if the pulses of F1 and F2 in the shift register become "high" (+ pulse) and the output signal of C1 becomes "high." Thus, C2 counts when the frame timing is regarded as correct. When C2 reaches a value of m (the number of stages of C2), it outputs a signal to reset C2 and C3.
- C3 counts the output signals of C1. Thus, C3 continues to count until it reaches a value of n (the number of stages of C3) even if the frame timing is incorrect. If C3 reaches a value of n, C2 and C3 are reset to zero and the frame synchronizing signal is renewed.

The frame synchronization system operates as follows.

- In the case of correct frame timing: When C1 reaches the full count of 2^K, it outputs the frame synchronizing signal. At that instant, the head chip of the code ought to stay at the position of F1 and F2 in the shift register. Then, C2 and C3 increase their counts by one. Here, we set $m < n$ because C2 can reach the full count before C3. When C2 reaches the full count, C2 and C3 are reset to zero by the resetting signal of C2 and the frame synchronizing signal is not renewed.
- In the case of incorrect frame timing: When the output of C1 becomes "high," F1 and F2 are not always the head chip. Thus, C3 tends to become full before C2. When C3 reaches the full count, C3 resets C2 and C3. Then the frame timing is renewed. The output signal of C3 makes C1 not count one chip. Thus, C1 can step back by a one-chip interval. In this way, the frame synchronizing signal is shifted. Thus, the frame timing is renewed.

III. SYNCHRONIZATION PERFORMANCE

In this section, we analyze performance of the frame synchronization system [7], [8]. We assume an AWGN environment and complete chip synchronization. Two general methods of initial acquisition exist: 1) transmitting a preamble and 2) self-synchronization, where the timing information is extracted from the received signal itself. We derive the initial acquisition time of the system without preamble for initial acquisition in Section III-A. Initial synchronization is acquired by the self-synchronization. We derive the initial acquisition time of the system with preamble in Section III-B. In Section III-C, we present retention time and recovery time.

A. Initial Acquisition Time Without Preamble

When C2 outputs the resetting signal, the frame timing is held. If the synchronization system maintains the frame timing, the probability P_h of holding frame timing is (Appendix A)

$$P_h(m,n,q) = \begin{cases} p^m, & \text{for } n-m=1 \\ p^m\left(1+\sum_{i=2}^{n-m}\sum_{j=1}^{[i/2]}\left\{\binom{m+j-1}{j}(pq)^j \cdot \binom{i-j-1}{i-2j}q^{i-2j}\right\}\right), & \text{for } n-m \geq 2 \end{cases} \tag{3}$$

where p is the probability of the pulse of the position of F2 getting +, q is the probability of the pulse of the position of F2 getting −, $(p+q=1)$ and $[\cdot]$ is Gauss notation.

When the frame timing is renewed consecutively, the probability P_r of renewing frame timing is (Appendix B)

$$P_r(m,n,q) = 1 - P_h(m,n,q) + pqP_h(m-1,n-1,q) - pqP_h(m,n-1,q). \quad (4)$$

When the frame timing is incorrect, the pulse of the position of F1 or F2 is equally likely to be + or −. Thus q is equal to 1/2. Therefore the probability P_R of renewing incorrect frame timing is given by

$$P_R(m,n) = P_r(m,n,1/2). \quad (5)$$

The probability $P_R(m,n)$ is independent of the code length.

After the frame timing renewal is carried out, it takes $T(m,n,q)$ frames until C2 reaches the full count of m (Appendix C). When the frame timing is held, it takes $T_{AA}(m,n,q)$ frames until C2 reaches the full count of m again (Appendix C). Then, it takes $T_h(m,n,q)$ frames as shown in (6) until the frame timing renewal is carried out (Appendix C)

$$T_h(m,n,q) = (1 - P_r(m,n,q)) \times \left(T(m,n,q) + T_{AA}(m,n,q)\frac{P_h(m,n,q)}{1-P_h(m,n,q)}\right) + n. \quad (6)$$

When the frame timing is incorrect, it takes $T_h(m,n,1/2)$ frames between frame timing renewals. If the frame synchronization is collapsed, $2^K - 1$ frames timing renewals are required in order to get back to correct frame timing. The chips which are skipped at the frame timing renewal amount to 2^K chips (= 1 frame). Thus, the number of frames which are required to get back to correct frame timing is given by

$$T_{r1}(m,n) = (2^K-1)\left\{T_h\left(m,n,\frac{1}{2}\right) + \frac{1}{2^K}\right\} + \frac{1}{2^K} = (2^K-1)T_h\left(m,n,\frac{1}{2}\right) + 1. \quad (7)$$

It is possible that framing pulses are not detected and the correct frame timing is renewed. Then, the system gets into a state of synchronization collapsed again. The average number of times of detection error is

$$D_e = \frac{P_r\left(m,n,\frac{1}{2}\mathrm{erfc}\left(\sqrt{\frac{K}{2^K}\frac{E_b}{N_0}}\right)\right)}{1 - P_r\left(m,n,\frac{1}{2}\mathrm{erfc}\left(\sqrt{\frac{K}{2^K}\frac{E_b}{N_0}}\right)\right)} \quad (8)$$

where E_b is the transmitted signal energy per message bit and N_0 is the noise power spectral density. Therefore, the initial acquisition time N_{acq1} is

$$N_{acq1} = \frac{1}{2}\left(T_{r1}(m,n) - \frac{1}{2^K}\right) + (T_{r1}(m,n)+n)D_e + T\left(m,n,\frac{1}{2}\mathrm{erfc}\left(\sqrt{\frac{K}{2^K}\frac{E_b}{N_0}}\right)\right). \quad (9)$$

The characteristics of initial acquisition time are shown in Fig. 2. The initial acquisition time can be made small by decreasing the value of n because renewing time is shorter.

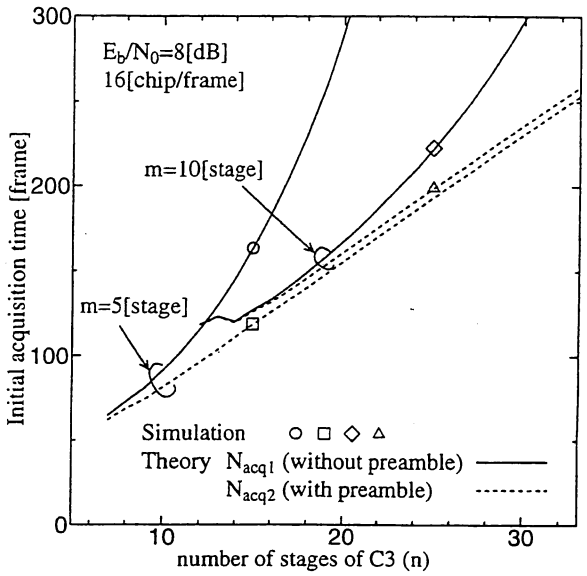

Fig. 2. Characteristics of initial acquisition time N_{acq1} and N_{acq2}, when E_b/N_0 = 8 dB, code length is 16 chips/frame, m (number of stages of C2) = 5 and 10 stages.

B. Initial Acquisition Time with Preamble

In this Section, we consider the system using a preamble code for acquisition only at the beginning of a series of transmissions. We derive the initial acquisition time when using preamble for initial acquisition. No information, other than preamble, is sent until the frame synchronization is established. Then following a switch to M-ary/SS system, information signal is transmitted. A code which consists of preamble has 2^K chips. The head chip of the code is + pulse and the other chips are all − pulses. From (7), the number of frames which are required to get back to correct frame timing is given by

$$T_{r2}(m,n) = (2^K-1)T_h\left(m,n,1-\frac{1}{2}\mathrm{erfc}\left(\sqrt{\frac{K}{2^K}\frac{E_b}{N_0}}\right)\right) + 1 \quad (10)$$

where $1 - \frac{1}{2}\mathrm{erfc}(\sqrt{\frac{K}{2^K}\frac{E_b}{N_0}})$ is the probability of the pulse of the position of F2 getting − under the incorrect frame timing. Therefore, the initial acquisition time N_{acq2} is

$$N_{acq2} = \frac{1}{2}\left(T_{r2}(m,n) - \frac{1}{2^K}\right) + (T_{r2}(m,n)+n)D_e + T\left(m,n,\frac{1}{2}\mathrm{erfc}\left(\sqrt{\frac{K}{2^K}\frac{E_b}{N_0}}\right)\right). \quad (11)$$

The characteristics of initial acquisition time are shown in Fig. 2. The frame synchronization system with preambles makes the acquisition time shorter than the system without preambles. Fig. 3 illustrates the characteristics of N_{acq2} when E_b/N_0 = 2, 5, and 8 dB. Under the condition of higher E_b/N_0, the initial acquisition time increases in proportion to the increase of the value of n. As the value of E_b/N_0 becomes high, the initial acquisition time gets short.

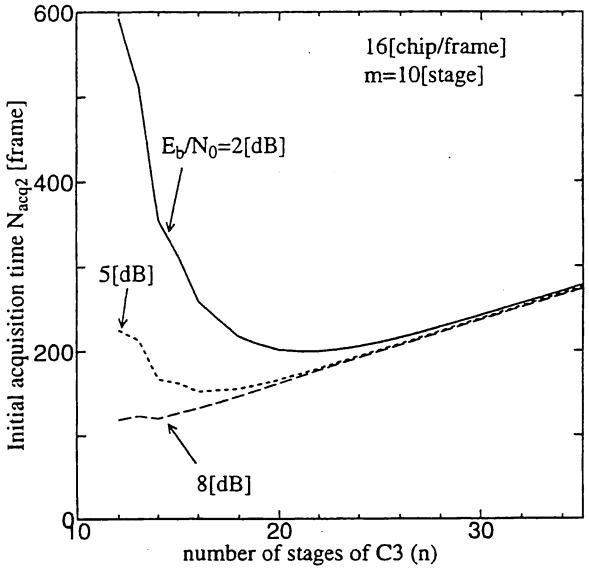

Fig. 3. Characteristics of initial acquisition time N_{acq2}, when $E_b/N_0 = 2$, 5, and 8 dB, code length is 16 chips/frame, m (number of stages of C2) = 10 stages.

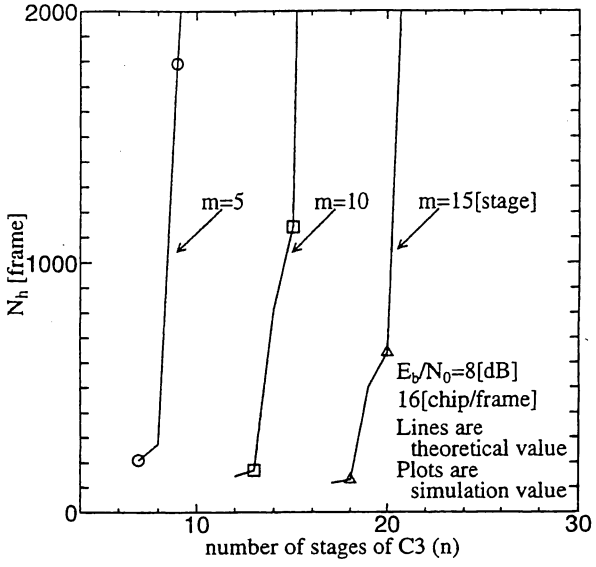

Fig. 4. Characteristics of N_h when code length is 16 chips/frame, $E_b/N_0 =$ 8 dB, and m (number of stages of C2)= 5, 10, and 15 stages.

C. Retention Time and Recovery Time

In this Section, we analyze performance of the frame synchronization system after the initial acquisition is accomplished. Based on the characteristics of T_h and T_{r1}, we calculate retention time (N_h) which is the average number of frames holding correct frame timing, and recovery time (N_r) which is the average number of frames required to establish synchronization after synchronization collapses.

When the frame timing is correct, q is

$$q = \frac{1}{2}\mathrm{erfc}\left(\sqrt{\frac{K}{2^K}\frac{E_b}{N_0}}\right). \tag{12}$$

The retention time N_h is obtained by substituting (12) in (6)

$$N_h = T_h\left(m, n, \frac{1}{2}\mathrm{erfc}\left(\sqrt{\frac{K}{2^K}\frac{E_b}{N_0}}\right)\right). \tag{13}$$

If the frame synchronization collapses, it takes $T_{r1}(m,n)$ frames to return to correct frame timing. Therefore, N_r is

$$N_r = T_{r1}(m,n) + (T_{r1}(m,n) + n)D_e + T\left(m, n, \frac{1}{2}\mathrm{erfc}\left(\sqrt{\frac{K}{2^K}\frac{E_b}{N_0}}\right)\right). \tag{14}$$

Fig. 4 shows the characteristics of N_h, where $E_b/N_0 = 8$ dB, 2^K (code length) = 16 chips/frame, and $m = 15$ stages. Retention time increases rapidly with increasing the value of n. Fig. 5 shows the characteristics of N_r. The rate of increase of the recovery time is smaller than that of the retention time. The recovery time is long when n is close to m because the probability of detection error increases. Characteristics from computer simulations are in good agreement with theoretical results in all cases for various m and n.

IV. Bit-Error Rate Performance

When the frame timing is correct, word error rate performance of this system is equal to that of the M-ary/SS communication system under complete synchronization. Let P_{w1} express this word error rate. On the other hand, when the frame timing is incorrect, the word error rate P_{w2} is $(2^K - 1)/2^K$. Thus, the word error rate P_w when considering frame synchronization is

$$P_w = \frac{N_h \cdot P_{w1} + \{T_{r1}(m,n) - 1\}P_{w2} + 1}{N_h + T_{r1}(m,n)} \tag{15}$$

where there is a wrong frame because the number of skipped chips is equivalent to a frame length, i.e., 2^K chips. BER P_b is

$$P_b = \frac{2^{K-1}}{2^K - 1}P_w. \tag{16}$$

Fig. 6 shows BER performance. If the number of stages of C3 increases, the curve gets close to the performance which is achieved under complete synchronization. Let $E_b/N_0 = 8$ dB, then the characteristics of BER versus the number of stages of C3 are shown in Fig. 7. Fig. 8 illustrates BER performance when the code length is 8, 16, and 32 chips/frame. It is possible to improve BER performance by increasing the number of stages of C3. For example, BER reaches $10^{-5.7}$ at 30 stages for C3 when C2 has 10 stages. Over 30 stages, there is little improvement.

V. Conclusion

We analyzed the performance of the M-ary/SS communication system using the Hadamard matrix and the racing counters. We discussed the following three points: 1) initial acquisition time of the system with preambles and the system without preambles, 2) retention time, and 3) recovery time. Moreover, we evaluated the BER performance by consid-

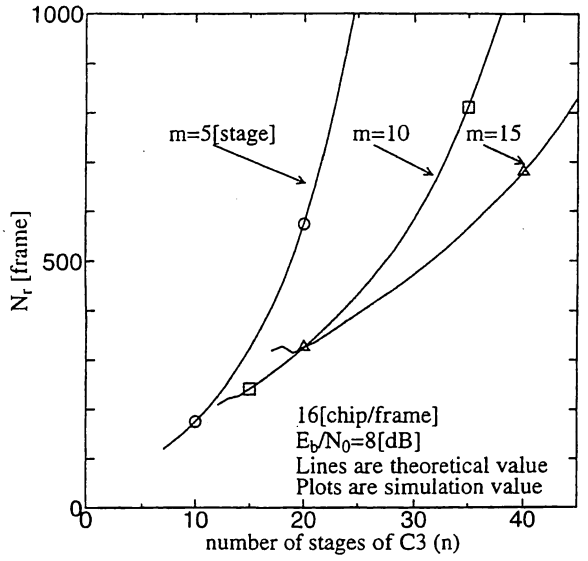

Fig. 5. Characteristics of N_r when code length is 16 chips/frame, E_b/N_0 = 8 dB, and m (number of stages of C2) = 5, 10, and 15 stages.

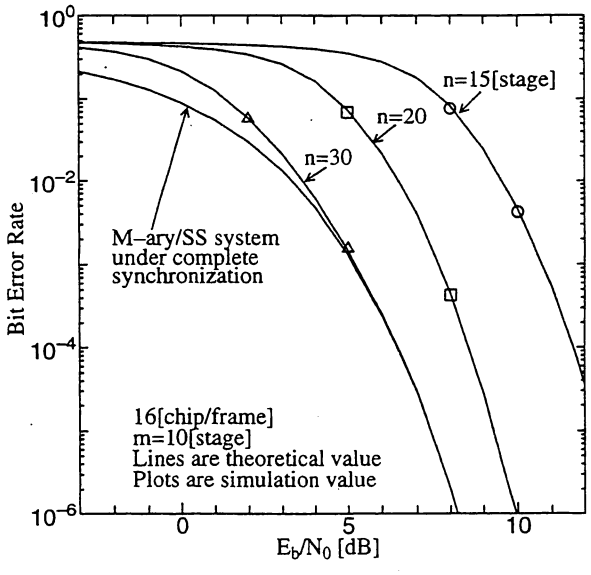

Fig. 6. BER performance when code length is 16 chips/frame, and m (number of stages of C2) = 10, and n (number of stages of C3) = 15, 20, and 30 stages.

ering frame synchronization performance. We obtained the following results.

- The frame synchronization system with preambles makes the acquisition time shorter than the system without preambles.
- We found that the retention time increases faster than recovery time with increasing the number of stages of C3.
- BER performance can be improved by increasing the number of stages of C3.
- $10^{-5.7}$ BER performance is achieved when the code length is 16 chips/frame, E_b/N_0 = 8 dB, m = 10 stages, and n = 30 stages.

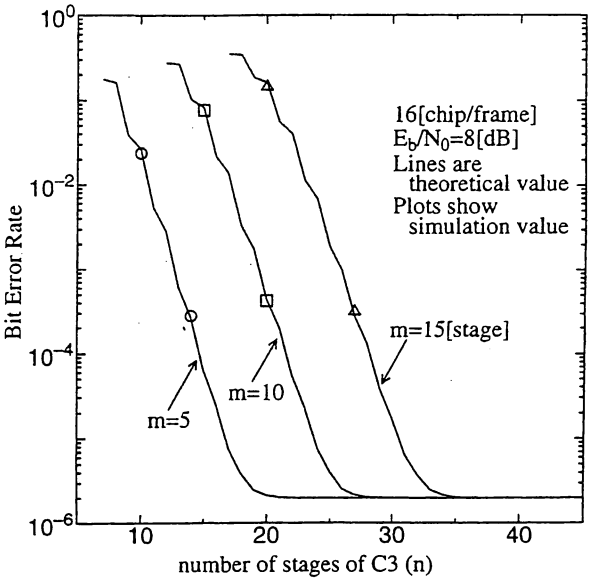

Fig. 7. BER versus the number of stages of C3 when code length is 16 chips/frame, E_b/N_0 = 8 dB, and m (number of stages of C2) = 5, 10, and 15 stages.

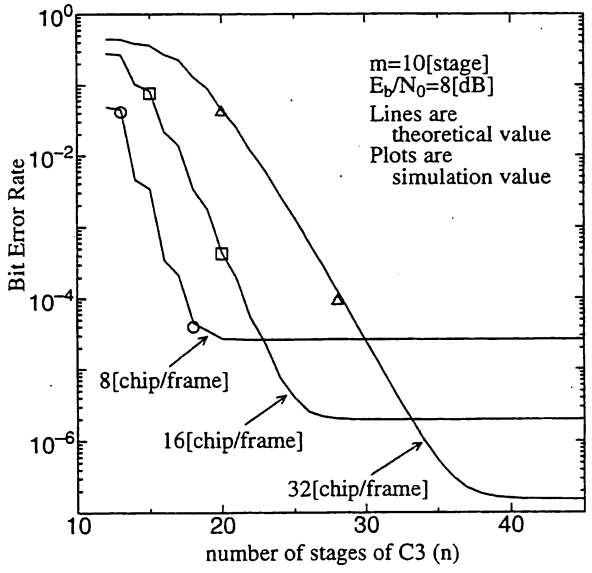

Fig. 8. BER versus the number of stages of C3 when code length is 16 chips/frame, E_b/N_0 = 8 dB, and m (number of stages of C2) = 10 stages.

Future work will include the analysis of chip synchronization, which was assumed as perfect in this paper.

APPENDIX

When C2 reaches the full count of m, operations of the frame synchronization system are expressed as a state transition diagram as shown in Fig. 9. The transition between the states depends on the pulse of the position of F2. The transition involves increasing both C2 and C3 or increasing only C2. It takes one frame interval to transit between the states.

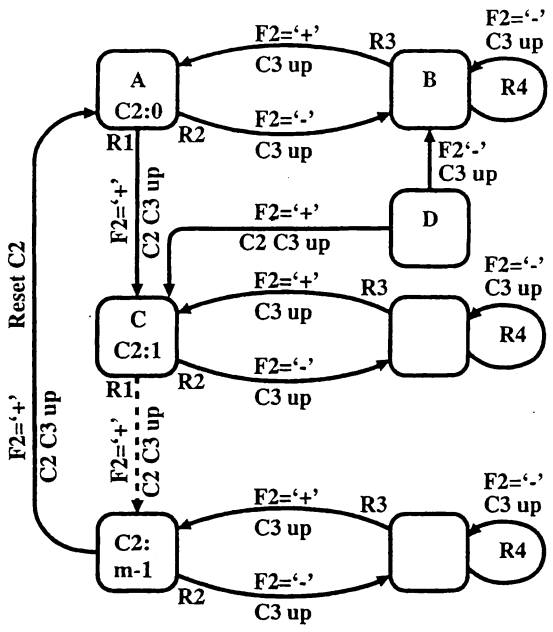

Fig. 9. State transition diagram of the synchronization system.

A. Derivation of P_h

If the transition starts from the state A, it is clear that it takes at least m transitions to fill the counter C2. The spare transitions, which are along the paths labeled R2, R3, and R4, are allowed $(n-m)$ times at most. If $i\,(\geq 2)$ spare transitions are performed, the paths labeled R2 are followed once at least and $[i/2]$ times at most, where $[\cdot]$ is Gauss notation. When j paths labeled R3 are assigned to m states (C1: $0, \cdots,$ C1: $m-1$), the repeated combination is

$$ {}_mH_j = \binom{m+j-1}{j}. \qquad (17) $$

When $(i-2j)$ paths labeled R4 are assigned to j paths labeled R3, the repeated combination is

$$ {}_jH_{i-2j} = \binom{i-j-1}{i-2j}. \qquad (18) $$

Therefore, the probability of C2 being full before C3 is

$$ P_h(m,n,q) = \begin{cases} p^m, & \text{for } n-m=1 \\ p^m\left(1+\displaystyle\sum_{i=2}^{n-m}\sum_{j=1}^{[i/2]}\left\{\binom{m+j-1}{j}(pq)^j \cdot \binom{i-j-1}{i-2j}q^{i-2j}\right\}\right), & \text{for } n-m\geq 2 \end{cases} \qquad (3) $$

where p is the probability of the pulse of the position of F2 being $+$, and q is the probability of the pulse of the position of F2 being $-$.

B. Derivation of P_r

When C3 reaches the full count of n, the frame timing renewal is carried out. The pulses of the position of F1 and F2 then become $(+,+)$, $(+,-)$,$(-,+)$, or $(-,-)$. If $(\mathrm{F1},\mathrm{F2}) = (+,+),$, the counters C2 and C3 increase, i.e., $(\mathrm{C2},\mathrm{C3}) = (1,1)$. In this case, the transition begins from the state C. The probability P_{CA} of returning to the state A from the state C is

$$ P_{\mathrm{CA}}(m,n,q) = P_h(m-1,n-1,q). \qquad (19) $$

If $(\mathrm{F1},\mathrm{F2}) = (-,+)$, C3 increases, i.e., $(\mathrm{C2},\mathrm{C3}) = (0,1)$. In this case, the transition begins from the state D. The probability P_{DA} of returning to the state A from the state D is

$$ P_{\mathrm{DA}}(m,n,q) = P_h(m,n-1,q). \qquad (20) $$

If $(\mathrm{F1},\mathrm{F2}) = (+,-)$ or $(-,-)$, C3 increase, i.e., $(\mathrm{C2},\mathrm{C3}) = (0,1)$. In this case, the transition begins from the state B. The probability P_{BA} of returning to the state A from the state B satisfies

$$ P_h(m,n,q) = pP_{\mathrm{CA}}(m,n,q) + qP_{\mathrm{BA}}(m,n,q). \qquad (21) $$

Therefore

$$ P_{\mathrm{BA}}(m,n,q) = \frac{P_h(m,n,q) - pP_h(m-1,n-1,q)}{q}. \qquad (22) $$

The probability of returning to the state A after the frame timing renewal is given by

$$ qP_{\mathrm{BA}}(m,n,q) + p^2P_{\mathrm{CA}}(m,n,q) + pqP_{\mathrm{DA}}(m,n,q). \qquad (23) $$

Thus, the probability of renewing frame timing consecutively is

$$ \begin{aligned} P_r(m,n,q) &= 1 - \{qP_{\mathrm{BA}}(m,n,q) + p^2P_{\mathrm{CA}}(m,n,q) \\ &\quad + pqP_{\mathrm{DA}}(m,n,q)\} \qquad (24) \\ &= 1 - P_h(m,n,q) + pqP_h(m-1,n-1,q) \\ &\quad - pqP_h(m,n-1,q). \qquad (4) \end{aligned} $$

C. Derivation of T_h

If the transition starts from the state A, it takes T_{AA} frames to return to the state A again. T_{AA} is given by

$$ T_{\mathrm{AA}}(m,n,q) = \begin{cases} m, & \text{for } n-m=1 \\ \frac{p^m}{P_h(m,n,q)}\left(m+\displaystyle\sum_{i=2}^{n-m}(m+i)\sum_{j=1}^{[i/2]}\left\{\binom{m+j-1}{j} \cdot (pq)^j\binom{i-j-1}{i-2j}q^{i-2j}\right\}\right), & \text{for } n-m\geq 2. \end{cases} \qquad (25) $$

If the transition starts from the state B, it takes T_{BA} frame to return to the state A. T_{BA} is given by

$$T_{BA}(m,n,q) = \frac{p^{m+1}}{P_{BA}(m,n,q)}\left(\sum_{i=0}^{n-m-2}(m+1+i) \times \sum_{j=0}^{[i/2]}\left\{\binom{m+j-1}{j}\cdot(pq)^j\binom{i-j}{i-2j}q^{i-2j}\right\}\right). \quad (26)$$

If the transition starts from the state C, it takes T_{CA} frames to return to the state A. T_{CA} is given by

$$T_{CA}(m,n,q) = T_{AA}(m-1,n-1,q). \quad (27)$$

If the transition starts from the state D, it takes T_{DA} frames to return to the state A. T_{DA} is given by

$$T_{DA}(m,n,q) = T_{AA}(m,n-1,q). \quad (28)$$

Therefore, the number of frames required to get to the state A after the frame timing renewal is

$$T(m,n,q) = 1 + q\cdot T_{BA}(m,n,q) + p^2\cdot T_{CA}(m,n,q) + pq\cdot T_{DA}(m,n,q). \quad (29)$$

The state A is repeated $P_h(m,n,q)/(1-P_h(m,n,q))$ times. Therefore, the number of frames required until the next frame renewal is given by

$$T_h(m,n,q) = (1-P_r(m,n,q)) \times \left(T(m,n,q) + T_{AA}(m,n,q)\frac{P_h(m,n,q)}{1-P_h(m,n,q)}\right) + n. \quad (6)$$

ACKNOWLEDGMENT

The authors would like to thank Professor M. Kobayashi of Ibaraki University.

REFERENCES

[1] P. K. Enge and D. V. Sarwate, "Spread spectrum multiple access performance of orthogonal codes: Linear receivers," *IEEE Trans. Commun.*, vol. COM-35, no. 12, pp. 1309–1319, Dec. 1987.
[2] A. Hammer and D. J. Schaefer, "Performance analysis of M-ary code shift keying in CDMA system," in *Int. Conf. Commun.*, Philadelphia, PA, June 1982.
[3] A. G. Burr, "M-ary bi-orthogonal keying for high rate digital radio-communication," in *Int. Conf. Data Commun. Technol. Proc.*, 1988, pp. 1–10.
[4] T. Yoshida and M. Nakagawa, "Spread spectrum synchronizing loop for spreading code modulated by code shift keying," *IECE Trans.*, vol. J67-B, no. 5, pp. 559–565, May 1984, (in Japanese).
[5] Y. Yano, H. Murai, H. Tachika, and T. Fujino, "A study on decision feedback code tracking loop for M-ary/SS communications," IEICE Tech. Rep., vol. SST92-63, Jan. 1993, (in Japanese).
[6] S. Tachikawa and G. Marubayashi, "A study of the performance of M-ary/spread spectrum communication systems using synchronous signal of PN sequence," IEICE Tech. Rep., vol. SSTA89-37, Nov. 1989, (in Japanese).
[7] K. Ohuchi, H. Habuchi, and T. Hasegawa, "A frame synchronization method using racing counters on M-ary orthogonal modulation systems," IEICE Tech. Rep., vol. SST94-51, Oct. 1994, (in Japanese).
[8] ——, "A study on the M-ary/SS communication system using a frame synchronization method of PCM communication systems," *IEICE Trans.*, vol. E77-A, no. 11, pp. 1942–1945, Nov. 1994.
[9] J. R. Gray and J. W. Pan, "Using digit statistics to word-frame PCM signals," *BSTJ*, vol. 43, no. 6, pp. 2985–3007, Nov. 1964.
[10] T. Sekimoto and H. Kaneko, "Group synchronization for digital transmission systems," *IRE Trans.*, vol. CS-10, no. 4, pp. 381–390, Dec. 1962.

Kouji Ohuchi was born in Ibaraki, Japan, in August 1970. He received the B.E. degree from Ibaraki University, Ibaraki, Japan, in 1994, where he is currently working toward the M.E. degree.

His research interests are in M-ary/SS communication systems and synchronization systems.

Mr. Ohuchi is a Student Member of IEICE of Japan.

Hiromasa Habuchi was born in Hyougo, Japan, on July 4, 1963. He received the B.E., M.E., and Ph.D. degrees in from Saitama University, Saitama, Japan, all in electrical engineering, in 1987, 1989, and 1992, respectively.

He joined in Faculty of Engineering, Ibaraki University, Ibaraki, Japan, in 1992, where he is a Research Associate. His research interests are in spread-spectrum communications, synchronization systems, and optical communications.

Dr. Habuchi is a member of IEICE of Japan.

Takaaki Hasegawa (S'82–M'86) received the B.E., M.E., and Ph.D. degrees from Keio University, Yokohama, Japan, all in electrical engineering, in 1981, 1983, and 1986, respectively.

He joined the Faculty of Engineering at Saitama University, Saitama, Japan, in 1986, where he has been an Associate Professor since 1991. His research interests include spread spectrum communications, human communications and human machine interfaces. He is the author of the book *Primary C Language Note* (Japanese, HBJ, 1989), a co-author of the books: *Fundamentals and Applications of Spread Spectrum Communication Technologies* (Japanese, Triceps, 1987), *Application Technologies of Spread Spectrum Communications* (Japanese, Triceps, 1992), and *Personal Communications and Consumer Communications* (Japanese, Baifukan, 1994).

Dr. Hasegawa is a member of IEICE and SITA.