



## Article

# Area-Efficient Universal Code Generator for GPS L1C and BDS B1C Signals

Jiwoon Park, Minsu Kim, Gwanghee Jo  and Hoyoung Yoo \* 

Department of Electronics Engineering, Chungnam National University, Daejeon 34134, Korea; jwpark.cas@gmail.com (J.P.); mskim.cas@gmail.com (M.K.); j\_ghee@cnu.ac.kr (G.J.)

\* Correspondence: hyyoo@cnu.ac.kr

**Abstract:** Recently, multi-frequency multi-constellation receivers have been actively studied, which are single receivers that process multiple global navigation satellite system (GNSS) signals for high accuracy and reliability. However, in order for a single receiver to process multiple GNSS signals, it requires as many code generators as the number of supported GNSS signals, and this is one of the problems that must be solved in implementing an efficient multi-frequency multi-constellation receiver. This paper proposes an area-efficient universal code generator that can support both GPS L1C signals and BDS B1C signals. The proposed architecture alleviates the area problem by sharing common hardware in a time-multiplex mode without degrading the overall system performance. According to the result of the synthesis using the CMOS 65 nm process, the proposed universal code generator has an area reduced by 98%, 93%, and 60% compared to the previous memory-based universal code generator (MB UCG), the Legendre-generation universal code generator (LG UCG), and the Weil-generation universal code generator (WG UCG), respectively. Furthermore, the proposed generator is applicable to all Legendre sequence-based codes.

**Keywords:** universal code generator; Legendre sequence; multi-constellation; GPS L1C; BDS B1C



**Citation:** Park, J.; Kim, M.; Jo, G.; Yoo, H. Area-Efficient Universal Code Generator for GPS L1C and BDS B1C Signals. *Electronics* **2021**, *10*, 2737. <https://doi.org/10.3390/electronics10222737>

Academic Editor: Kiat Seng Yeo

Received: 23 September 2021  
Accepted: 8 November 2021  
Published: 10 November 2021

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

A global navigation satellite system (GNSS) calculates navigation using constellation satellites and provides users with global-level location and time information [1,2]. GNSS receivers distinguish visible satellites and extract navigation messages from mixed signals coming from many satellites. In this case, the pseudo-random noise (PRN) codes included in the satellite signals play an important role [2,3]. Mathematically, codes in which 0s and 1s are randomly well-distributed have the characteristic of having high auto-correlations and low cross-correlations [2,3]. Navigation systems extract the signal information of a certain satellite from mixed signals coming from many satellites using PRN codes with such a correlation characteristic [2,3]. Each satellite combines a unique PRN code with navigation data and transmits the resultant signals, and the receiver receives signals transmitted from all the satellites in a mixed form. The receiver sequentially calculates the correlation values between the candidate PRN codes generated internally with the mixed signals transmitted by all the satellites. If the correlation value between the signals received by the receiver and the generated PRN code is high, the satellite corresponding to the currently generated PRN code will be judged to be a visible satellite, and, if the correlation value is low, the satellite corresponding to the currently generated PRN code will be judged to not be included in the satellites that transmitted the signals currently received. For instance, Figure 1 depicts the general signal acquisition for a GNSS receiver. When PRN 2, 3, and 4 are visible satellites, the GNSS receiver takes the mixed signals coming from PRN 2, 3, and 4. The GNSS receiver computes the correlation between the received signal and the internally generated PRN code. In this example, the generated PRN 2, 3, and 4 has a high correlation, but the generated PRN 1, 5, 6, 7, and 8 maintain a low correlation. Therefore, the types of satellites included among those that transmitted the signals currently received

can be judged by using the correlation characteristic of PRN codes, and this is an essential calculation process for performing navigational calculations in the satellite navigation system [3].

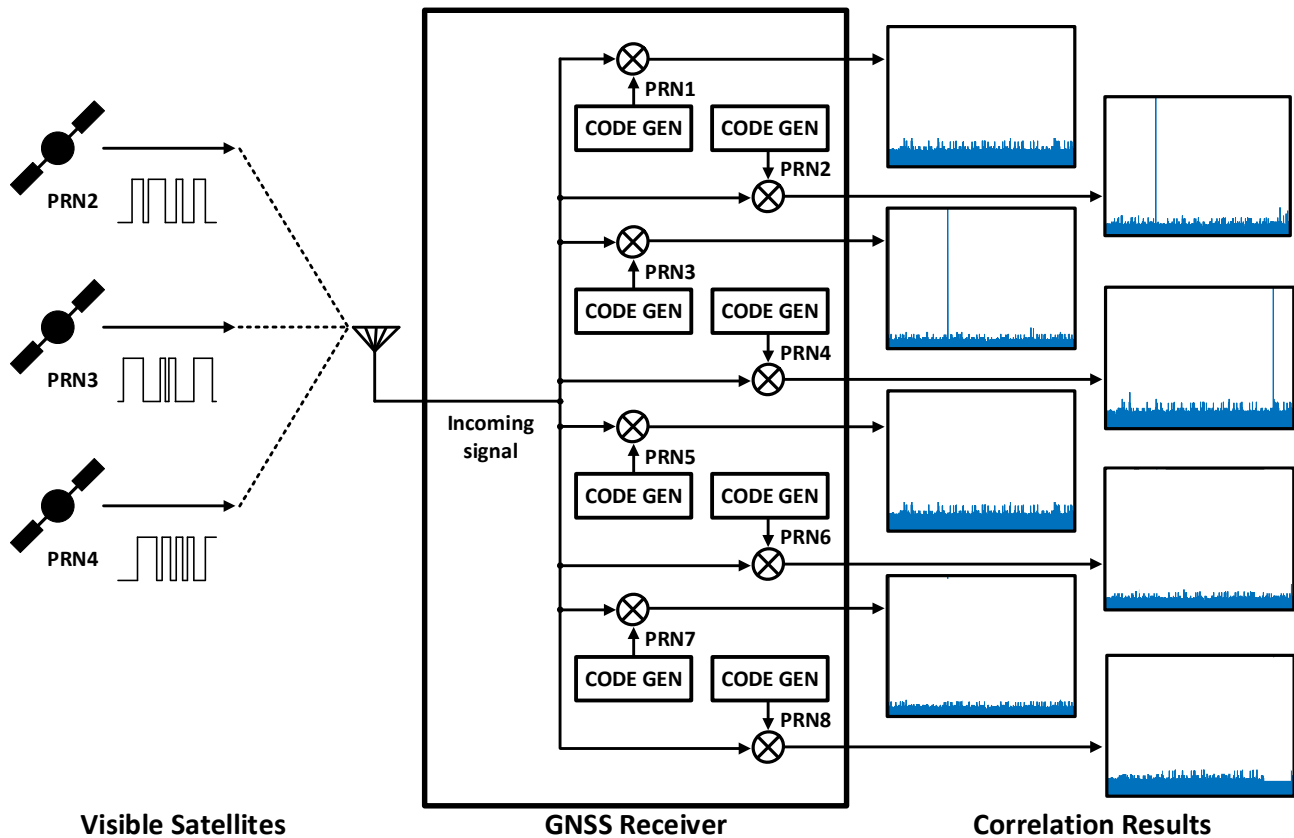


Figure 1. Signal acquisition process for a GNSS receiver.

PRN codes have unique PRN values according to satellites' signal systems and numbers [4–7]. Representative examples of satellite signal systems that provide the GNSS include GPS [4] in the United States, GALILEO [5] in the EU, GLONASS [6] in Russia, and BDS [7] in China. Each satellite signal system transmits many satellite signals in various frequency bands. More precisely, GPS [4] satellites transmit L1 C/A, L1C, L2C, and L5 signals, and GALILEO [5] satellites transmit E1 and E5a/b signals. GLONASS [6] satellites transmit L1OF, L1OC, L2OF, L2OC, and L3OC signals, and BDS [7] satellites transmit B1I, B1C, B2I, B2A, and B3I signals. Individual satellites in each satellite signaling system have their unique PRN codes to achieve high auto-correlation and low cross-correlation values, which are described in the interface control document (ICD) of each signaling system [4–7]. Table 1 shows a summary of the PRN code generation methods by satellite system signal. Code generation methods are generally divided into three types: LFSR-based codes, memory-based codes, and Legendre sequence-based codes. LFSR-based codes have the advantage that maximally long PRN codes can be generated with simple LFSR hardware, and most satellite signals are made using LFSR-based codes. In the case of memory-based codes, mathematically pre-calculated PRN code sequences are stored in the memory, and the sequences are read from the memory and used as PRN codes. Memory-based codes are used in Galileo E1 signals because they are intuitive and simple, but they have the disadvantage that a lot of memory space is necessary to store the sequences. Lastly, Legendre sequence-based codes show excellent correlation characteristics, and they have recently been used in GPS L1C and BDS B1C signals.

**Table 1.** Summary of GNSS signals.

GNSS	Frequency Band <sup>a</sup>	Signal	Code Structure
GPS [4]	L1	L1C/A	LFSR
		L1C	Legendre sequence
	L2	L2C (L2CM/L2CL)	LFSR
	L5	L5 I	LFSR
		L5 Q	LFSR
GALILEO [5]	E1	E1	Memory code
	E5	E5a (E5a-I/E5a-Q)	LFSR
		E5b (E5b-I/E5b-Q)	LFSR
GLONASS [6]	L1	L1OF	LFSR
		L1OC (L1OCd/L1OCp)	LFSR
	L2	L2OF	LFSR
		L2OC (L2OCp)	LFSR
	L3	L3OC (L3OCd/L3OCp)	LFSR
	BDS [7]	B1	B1I
L1		B1C	Legendre sequence
		B2I	LFSR
L5		B2A (B2ad/B2ap)	LFSR
B3		B3I	LFSR

<sup>a</sup> L1:1575.42 MHz, L2: 1227.60 MHz, L3: 1381.05 MHz L5: 1176.45 MHz; B1:1561.098 MHz, B2: 1207.140 MHz, B3:1268.520 MHz; E1: 1575.420 MHz, E5: 1191.795 MHz.

As the fields of the application of the GNSS have expanded along with the recent development of the industries, studies to provide high precision and additional functions of GNSS receivers have been extensively conducted [8–17]. In addition to the simple role of receiving signals and performing navigation calculations, advanced signal technologies, such as anti-jamming [8] and anti-spoofing [9], are applied, and studies on multi-constellation and multi-frequency receivers [10–12] are being conducted for high-precision high-reliability location calculations. Especially, studies on single code generators that support various codes are actively conducted for the implementation of multi-constellation and multi-frequency receivers so that single receivers can support multiple codes [13–17]. As shown in Table 1, since LFSR-based code generation is the most widely applied, studies on LFSR-based codes have been steadily conducted [10–13]. According to one study [13], many LFSR-based PRN codes can be generated using a single LFSR structure. However, there are not so many studies on the Legendre-based code generator structure so far. As representative Legendre-based generator structures, authors of [14,15] proposed general code generators that use memory to store and read Legendre-based codes. The authors of [16] proposed a code generator structure where Legendre sequences are generated on the fly to generate PRN codes and to the authors of [17] proposed a Weil-generation code generator structure where Legendre sequences are stored in the ROM and Weil sequences are generated on the fly. Although previous studies [14–17] succeeded in generating multiple Legendre-based codes using single hardware, they still use a lot of unnecessary hardware resources to universally generate Legendre-based codes. Therefore, this paper proposes an area-efficient code generator structure where many Legendre sequence-based codes can be generated using a single universal code generator by removing redundant used hardware resources within a range where the overall system performance is not degraded. The rest of this paper is organized as follows. Section 2 explains the background of Legendre sequence-based PRN codes, and Section 3 describes the previous universal code generators. The proposed universal code generator is proposed in Section 4, the experimental results are discussed in Section 5, and the concluding remarks are made in Section 6.

## 2. Background

To analyze the structure of code generators, it is necessary to fully understand signal configuration and code construction. In this section, signal configuration and code con-

struction are mathematically described to clarify the structure of code generators in the following sections.

### 2.1. Signal Configuration

In general, each satellite signal that includes Legendre-based codes includes a data channel and a pilot channel, as shown in Equation (1). The data channel delivers navigation messages, and the pilot channel plays the role of facilitating signal acquisition and tracking [4,7].

$$S(t) = S_{data}(t) + jS_{pilot}(t) \quad (1)$$

where  $S_{data}(t)$  denotes the data component, which is an in-phase component of signals,  $j$  is an imaginary unit, and  $S_{pilot}(t)$  denotes the pilot component, which is a quadrature component of signals. The data component  $S_{data}(t)$  is generated as a PRN code  $C_{data}(t)$  modulated with a navigation message  $D_{data}(t)$  and a subcarrier  $sc_{data}(t)$  for the data component, and the pilot component  $S_{pilot}(t)$  is generated from the PRN code  $C_{pilot}(t)$  modulated with a subcarrier  $sc_{pilot}(t)$  for the pilot component. If the power ratio of the two components is assumed to be 1:3, the two components can be expressed as shown by Equation (2). Table 2 is a summary of detailed information on the Legendre sequence-based GPS L1C [4] signals and BDS B1C [7] signals.

$$\begin{aligned} S_{data}(t) &= \frac{1}{2}D_{data}(t)C_{data}(t)sc_{data}(t) \\ S_{pilot}(t) &= \frac{\sqrt{3}}{2}C_{pilot}(t)sc_{pilot}(t) \end{aligned} \quad (2)$$

**Table 2.** GPS L1C and BDS B1C signals.

Signal Property	GPS L1C	BDS B1C
Frequency (MHz)	1575.42 (L1)	1575.42 (L1)
Modulation	TMBOC	QMBOC
Data code ( $C_{data}$ )	Plain code	Plain code
Pilot code ( $C_{pilot}$ )	Tiered code	Tiered code
Code rate (Mcps)	1.023	1.023

### 2.2. Legendre Sequence-Based PRN Code Generation

To enable receivers to distinguish satellite signals, each satellite transmits signals with individual PRN codes, such as  $C_{data}(t)$  and  $C_{pilot}(t)$ , shown in Equation (2). The PRN codes are divided according to the configurations into plain codes consisting of only primary codes and tiered codes in which primary codes and secondary codes are combined.

The generation method is the same for both primary codes and secondary codes. Firstly, a Legendre sequence is generated; next, a Weil sequence is configured based on the Legendre sequence, and a primary code or a secondary code is generated through the post-processing of the Weil sequence. In the case of a plain code, the primary code becomes the final PRN code, and, in the case of a tiered code, the final PRN code is determined by the combination of the primary code and the secondary code.

To examine the foregoing in terms of formulas, first, a Legendre sequence, the length of which is  $N_L$ , is defined as shown in Equation (3).

$$L(k) = \begin{cases} 0, & k = 0 \\ 1, & k \neq 0 \text{ and } k = x^2 \bmod N_L \text{ (integer } x) \\ 0, & \text{else} \end{cases} \quad (3)$$

where  $k$  is the index of Legendre sequence for  $(0 \leq k < N_L)$ . Then, a Weil sequence, of which the length is  $N_W$ , is calculated through the modulo-2 addition operation of two Legendre sequences with a phase difference as shown in Equation (4).

$$W(k; w) = L(k) \oplus L(k + w) \quad (4)$$

where  $k$  is the index of Weil sequence for ( $0 \leq k < N_W$ ) and  $\omega$  is the phase difference between two Legendre sequences. Finally, a post-processing process is carried out to generate a primary code or a secondary code, of which the length is  $N_C$ , using the Weil sequence expressed as shown in Equation (4). In the case of GPS L1C signals, a post-processing process through which a 7-bit fixed sequence  $\{0,1,1,0,1,0,0\}$  is inserted based on the insertion point  $\tau$  is carried out.

$$C(k;w;\tau) = \begin{cases} 0, & k = \tau - 1, \tau + 2, \tau + 4, \tau + 5 \\ 1, & k = \tau, \tau + 1, \tau + 3 \\ W(k;\omega), & \text{else} \end{cases} \quad (5)$$

where  $k$  is the index of GPS L1C code for ( $0 \leq k < N_C$ ) and  $\tau$  is the insertion point. Every GPS satellite in a GPS satellite system has a unique phase difference  $\omega$  and an insertion point  $\tau$  and generates a unique code through the phase difference and insertion point. In addition, in the case of BDS B1C signals, a post-processing process is carried out through which the Weil code, of which the length is  $N_W$ , is periodically truncated as shown in Equation (6).

$$C(k;w;\tau) = W((k + p - 1) \bmod N_W; \omega) \quad (6)$$

where  $k$  is the index of BDS B1C code for ( $0 \leq k < N_C$ ) and  $p$  is the truncation point. Furthermore,  $p$  indicates the Weil code truncation point while meaning that the code starts from the  $p$ th Weil sequence chip. BDS B1C satellites in the BDS satellite system generate individual codes by using their unique phase difference  $\omega$  and truncation point  $p$ . Table 3 summarizes the lengths and detailed parameters by sequence of GPS L1C signals and BDS B1C signals.

**Table 3.** Code parameters for GPS L1C and BDS B1C signals.

Code Property	GPS L1C			BDS B1C		
	Data	Pilot		Data	Pilot	
Channel Code Type	Primary Legendre	Primary Legendre	Secondary LFSR	Primary Legendre	Primary Legendre	Secondary Legendre
Code Length $N_C$	10,230	10,230	2047	10,230	10,230	1800
Weil Length $N_W$	10,223	10,223	-	10,243	10,243	3607
Legendre Length $N_L$	10,223	10,223	-	10,243	10,243	3607
Phase difference $w$	1–5111	1–5111	-	1–5121	1–5121	1–1803
Insertion Point $\tau$	1–10,223	1–10,223	-	-	-	-
Truncation Point $p$	-	-	-	1–10,243	1–10,243	1–3607

### 3. Previous Code Generator Structures

As described in Table 3, each satellite has a unique phase difference  $\omega$ , an insertion point  $\tau$ , and a truncation point  $p$ , and generates a unique PRN code according to Equations (3)–(6). In order to implement the GNSS receiver in hardware, the implementation of a code generator is essential. Whereas existing studies on code generators were conducted focusing on LFSR-based codes [10–13], a few studies on Legendre sequence-based codes have recently been conducted [14–17]. Figures 2–4 show the structure of Legendre sequence-based code generators. Figure 2 shows the structure of memory-based universal code generators (MB UCG) [14,15], Figure 3 shows the structure of Legendre-generation universal code generators (LG UCG) [16], and Figure 4 shows the structure of Weil-generation universal code generators (WG UCG) [17]. The memory-based universal code generator [14,15] shown in Figure 2 has a structure where the final PRN codes generated according to Equations (3)–(6) are stored in the memory for its operation.

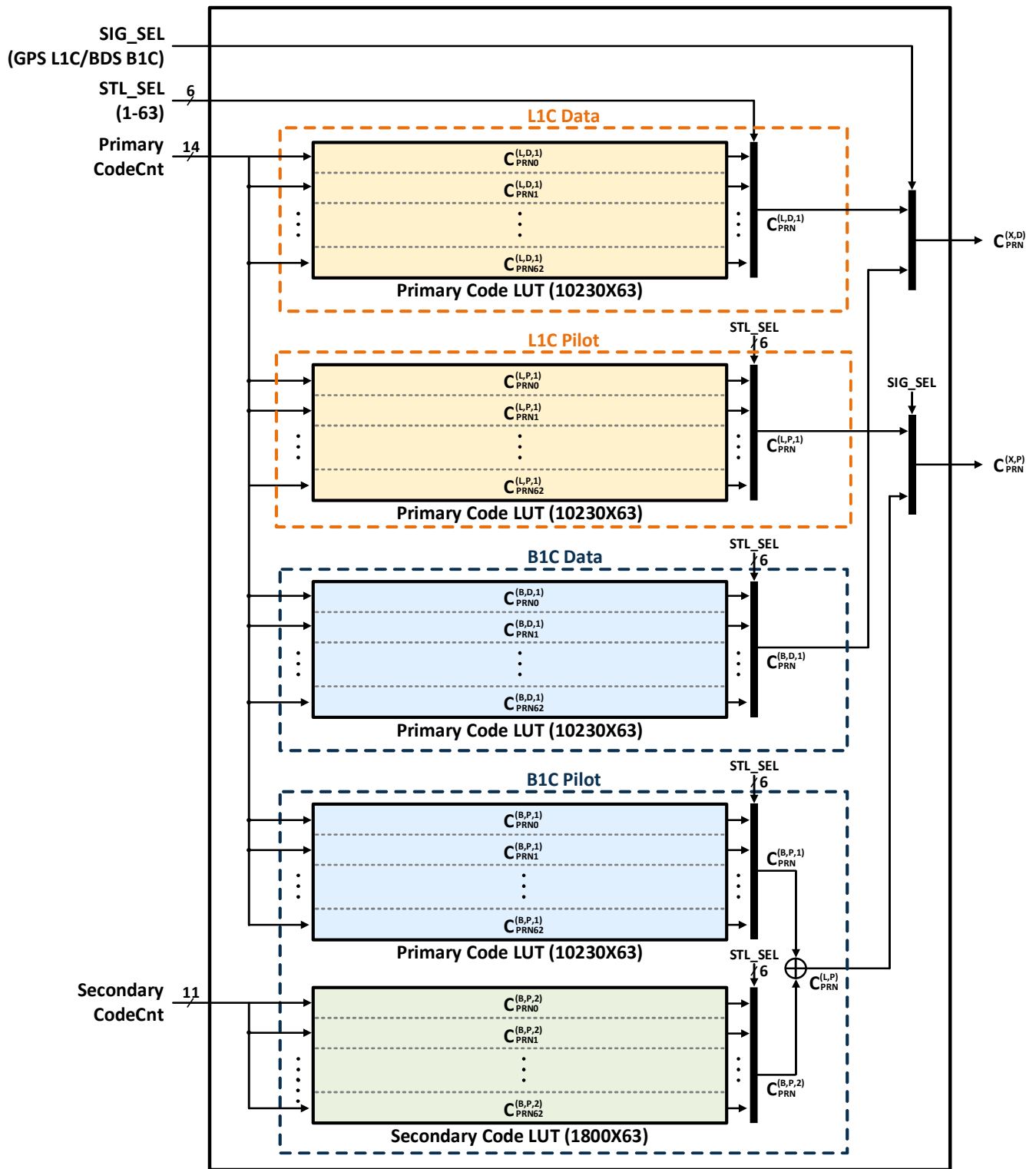


Figure 2. Memory-based universal code generator [14,15] for GPS L1C and BDS B1C.

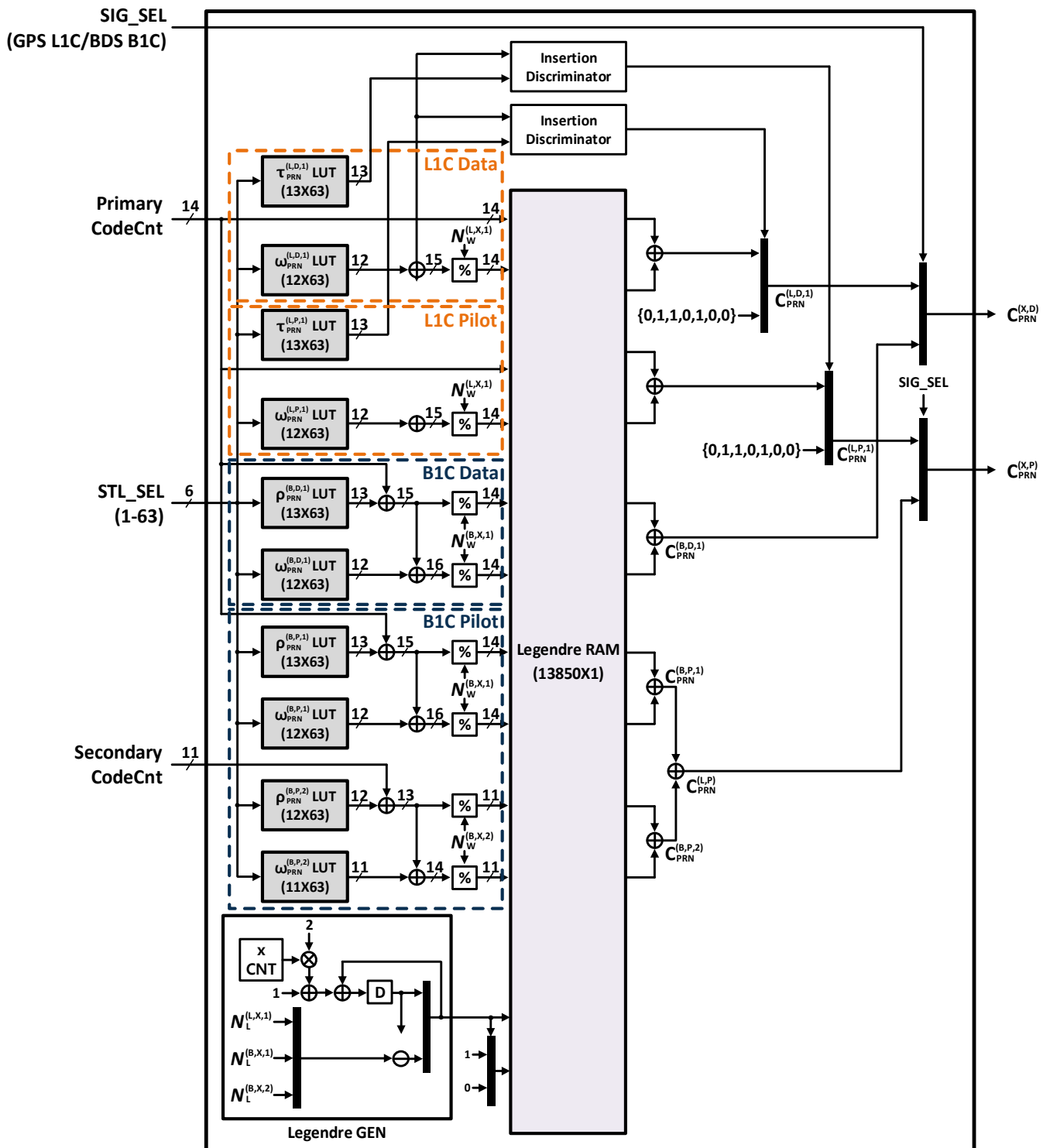


Figure 3. Legendre-generation universal code generator [16] for GPS L1C and BDS B1C.

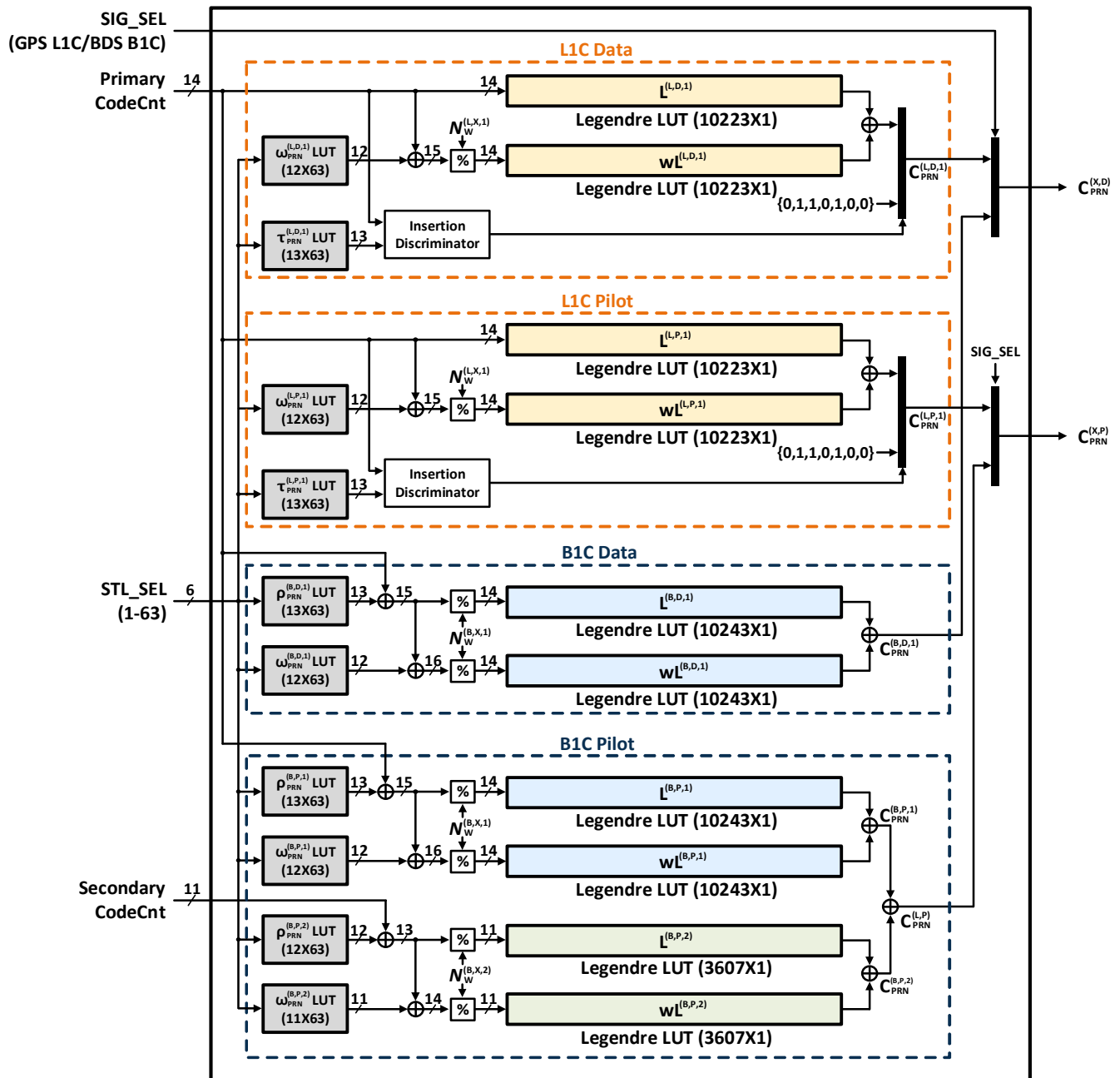


Figure 4. Weil-generation universal code generator [17] for GPS L1C and BDS B1C.

To operate as a universal code generator, the memory-based universal code generator calculates all the PRN sequences by signal (GPS L1C, BDS B1C), by channel (Data, Pilot), and by satellite (1~63), stores the calculated PRN sequences in the ROM, and outputs the bits of the given PRN code sequence according to external control signals. The notations of detailed signals are summarized in Table 4. As shown in Figure 2, when a GPS L1C or BDS B1C signal has been selected by SIG\_SEL and the satellite number has been input by STL\_SEL, the data channel code  $C_{data}(t)$  and pilot channel code  $C_{pilot}(t)$  are selectively generated.



**Table 4.** Notations.

Notation	Meaning
$(L, -, -)$	GPS L1C signal
$(B, -, -)$	BDS B1C signal
$(X, -, -)$	Universal signals
$(-, P, -)$	Pilot channel
$(-, D, -)$	Data channel
$(-, X, -)$	All channels
$(-, -, 1)$	Primary code
$(-, -, 2)$	Secondary code
$(-, -, 3)$	All codes

Although memory-based universal code generators [14,15] can select the GNSS signals, channels, and satellites according to control signals to generate desired codes, the required memory size is very large because they store the codes for all the cases. To supplement the foregoing, a structure to generate Legendre sequences on the fly according to Equation (3) and a structure to generate Weil sequences on the fly according to Equation (4) were proposed. Whereas memory-based universal code generators [14,15] store entire codes by signal (GPS L1C, BDS B1C), by channel (Data, Pilot), and by satellite (1~63) in the ROM in advance, Legendre-generation universal code generators (LG UCG) [16] store the phase differences  $\omega$ , insertion points  $\tau$ , and truncation points  $p$  required by signal (GPS L1C, BDS B1C), by channel (Data, Pilot), and by satellite (1~63) in the ROM. According to Figure 3, the parameters are selected from the ROM according to the given inputs SIG\_SEL and STL\_SEL, and the operation for Equation (3) is carried out on the fly using the Legendre-GEN circuit. The generated Legendre sequences are stored in the RAM, and the data channel code  $C_{data}(t)$  and the pilot channel code  $C_{pilot}(t)$  are generated through the calculation process shown in Equations (4)–(6).

Finally, Figure 4 shows the architecture of Weil-generation universal code generators (WG UCG) [17]. Legendre-generation universal code generators [16] use a small ROM but require RAM to store the Legendre sequences generated on the fly. In general, RAM, which simultaneously supports read and write functions, requires more hardware resources and has longer latency compared to ROM, which provides only a simple read function. In addition, whereas ROM does not require the initialization time to generate the initial value, the initialization time is essential in the case of RAM because it can perform reading only after writing the initial value first. In the case of the Weil-generation universal code generators [16] shown in Figure 4, the RAM in the Legendre-generation universal code generator shown in Figure 3 [15] was completely removed. Instead of removing the RAM memory, the Legendre ROM memory sequences by signal (GPS L1C, BDS B1C) and by channel (Data, Pilot) are added, and the Weil sequences are generated on the fly according to Equation (4). Through the foregoing, the complexity of the RAM memory is reduced, while the time required for initialization is completely eliminated.

#### 4. Proposed Area-Efficient Universal Code Generator

Although previous studies [14–17] succeeded in generating multiple Legendre-based codes using single hardware, the relevant code generators still have high hardware complexity to universally generate Legendre-based codes. Concretely, the memory-based universal code generator [14,15] shown in Figure 2 requires a total of 2.7 Mb of ROM to store all the PRN codes by signal (GPS L1C, BDS B1C), by channel (Data, Pilot), and by satellite (1~63). The Legendre-generation universal code generator [16] shown in Figure 3 uses 8.1 K of ROM to store the parameters by signal (GPS L1C, BDS B1C), by channel (Data, Pilot), and by satellite (1~63), and additionally requires 13.8 Kb of RAM to store

the generated Legendre sequences and 6.9 K clock cycle initialization time to generate Legendre sequences. The Weil-generation universal code generator [17] shown in Figure 4 requires 97.2 Kb of ROM as read-only memory to store the parameters and Legendre sequences by signal (GPS L1C, BDS B1C), by channel (Data, Pilot), and by satellite (1~63).

In addition, since previous studies [14–17] carried out designs without considering the entire GNSS receiver system, designs that consider the operation elements of the entire system are necessary. All the previous universal code generators [14–17] generate a data channel code  $C_{data}(t)$  and a pilot channel code  $C_{pilot}(t)$  for every clock cycle. In general, if it is assumed that the GNSS receiver operates at several hundred MHz, the code rate of the previous universal code generator will reach several Mbps. Although a higher code rate can support a higher-speed receiver, when seen from the viewpoint of the system, several Mbps is an unnecessarily high value. According to Table 1, since the code rate of GPS L1C signals and BDS B1C signals is 1.023 Mbps, if the code generator can provide a throughput of 1.023 Mbps or more, there is no degradation from the viewpoint of the entire system.

Therefore, in this paper, the hardware complexity will be reduced by sharing the common hardware within the range where there is no degradation from the viewpoint of the system. Time multiplexing is a useful scheme of utilizing shared hardware. It is a method of using shared hardware according to the scheduling of a time unit instead of using the same hardware at the same time by duplicating the same hardware. Time multiplexing can lower the hardware complexity at the cost of increasing latency. Additionally, it is usually implemented using simple logics, such as a multiplexer, and a control signal that changes in every time unit. The proposed code generator also applies time multiplexing with simple logics. Weil-generation universal code generators [17] that correspond to the state of the art use two Legendre ROMs for each channel (Data, Pilot) to generate Weil codes on the fly according to Equation (4). In this case, each Legendre ROM provides two Legendre sequences with a phase difference to configure a Weil sequence, as shown in Equation (4).

However, since there will be no degradation in the entire system if only the code generator operates at least at 1.023 Mbps, in the proposed code generator structure, the Legendre ROMs existing by channel are shared, and the Weil sequence shown in Equation (4) is calculated through time multiplexing. Furthermore, as seen in Table 2, in the case of GPS L1C signals, the Legendre sequence of the data channel and the Legendre sequence of the pilot channel are the same. Therefore, in the case of the GPS L1C signals, the data channel and the pilot channel can share the Legendre ROM. In conclusion, the proposed universal code generator where the Legendre ROMs and associated circuitry in the structure of the Weil-generation universal code generator shown in Figure 4 [17] are shared can be seen in Figure 5. Unlike the structure of the Weil-generation universal code generator with four 10,223 Legendre ROMs, two 10,243 Legendre ROMs, and two 3607 Legendre ROMs, the proposed universal code generator has only one each of the 10,223 Legendre ROM, 10,243 Legendre ROM, and 3607 Legendre ROM, and the complexity is reduced as these Legendre ROMs are shared. Although the latency increases due to the flip-flop and time-multiplexing to store the intermediate Legendre value as a trade-off, there is no degradation because the throughput exceeds 1.023 Mbps at operating frequencies not lower than the general MHz. In the case of the proposed universal code generator shown in Figure 5, 3.2 Kb of ROM is required with no RAM, and the latency increases by six clock cycles at the maximum for time multiplexing.

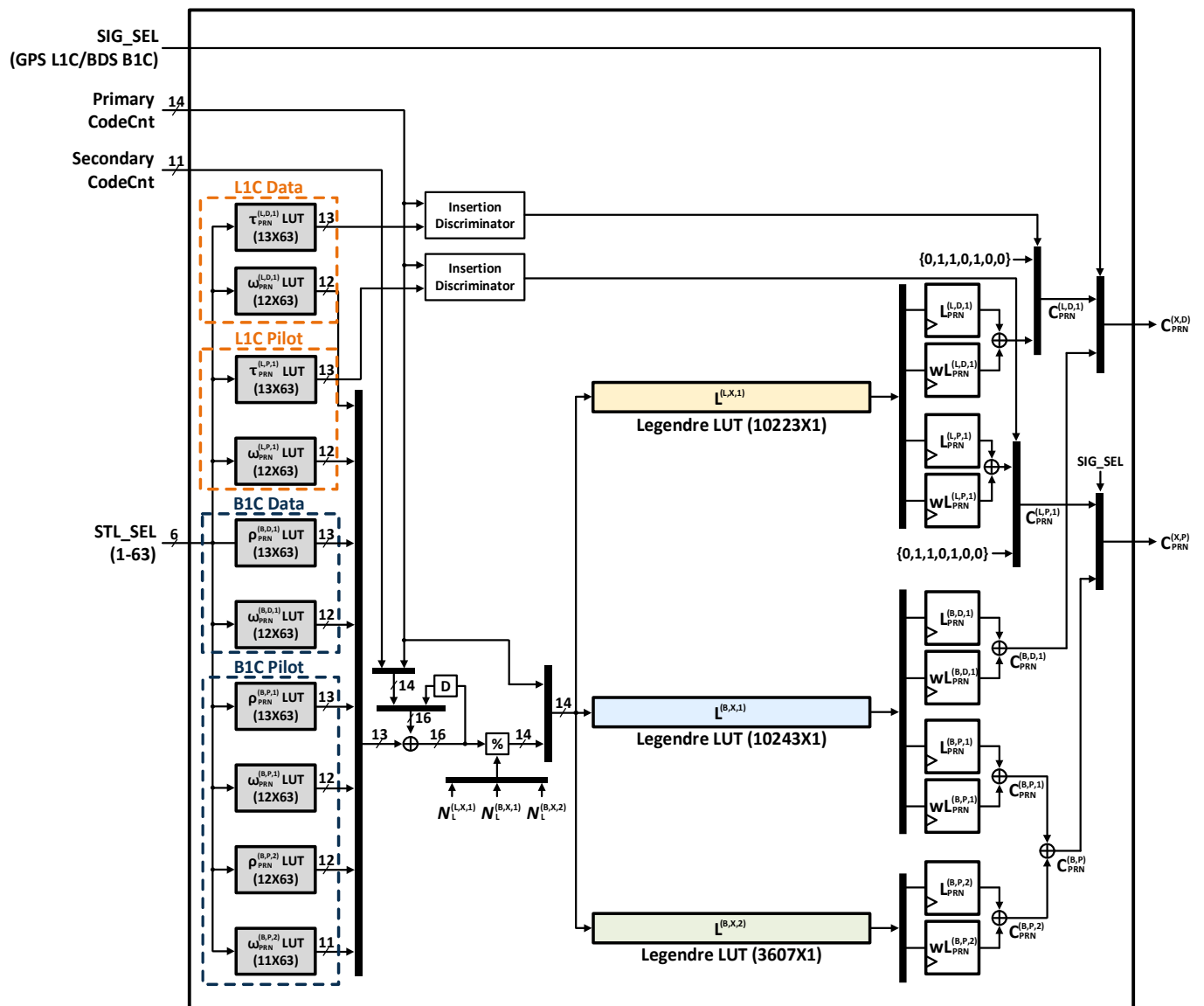


Figure 5. Proposed universal code generator.

### 5. Experimental Results

Using the CMOS 65 nm process, the proposed universal code generator, a memory-based universal code generator (MB UCG) [14,15], a Legendre-generation universal code generator (LG UCG) [16], and a Weil-generation universal code generator (WG UCG) [17] were implemented using Verilog HDL and simulated with Cadence NC-Verilog 13.1.

To verify valid code generation, all the codes generated from the proposed universal code generator, MB UCG, LG UCG, and WG UCG are compared with the code provided from the official ICD. According to the experiments, all the codes generated from code generators and provided from the official ICD are the same. For instance, Figure 6 shows the snapshot of the first and last 24 chips for the BDS B1C codes of PRN1, and each of the 24 chips are the same as those from the official BDS B1C ICD [7]. As a result, we verified that all the codes are correctly computed, and it implies that all the codes can be used for acquisition and tracking in a GNSS receiver.

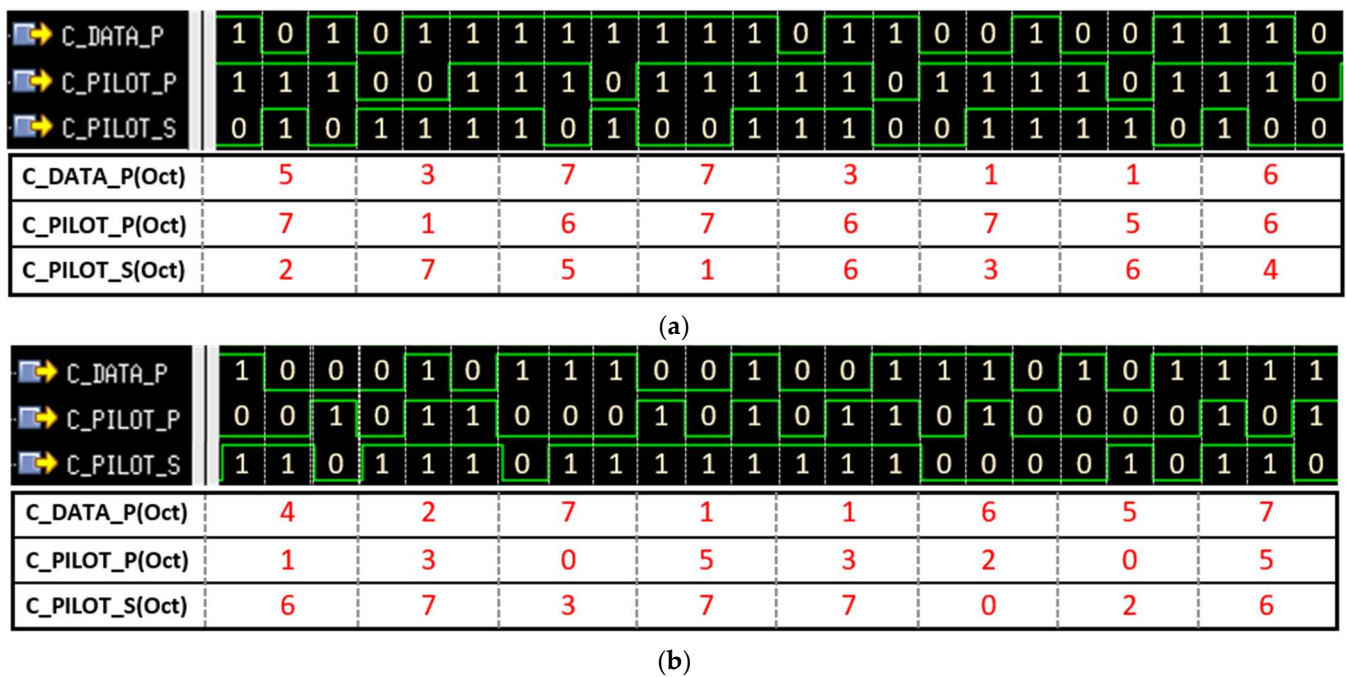


Figure 6. Simulation results. (a) First 24 chips for BDS B1C codes of PRN1. (b) Last 24 chips for BDS B1C codes of PRN1.

Table 5 summarizes the results of the synthesis using Synopsys Design Compiler 2015.06-SP2 with the operating frequency 200MHz. Since the memory-based universal code generator [14,15] stores the codes by signal (GPS L1C, BDS B1C), by channel (Data, Pilot), and by satellite (1~63) in the ROM, it has a 925K equivalent gate count, which is the largest area among the universal code generators. The Legendre-generation universal code generator [16] reduced the size of the entire ROM by storing the parameters by signal (GPS L1C, BDS B1C), by channel (Data, Pilot), and by satellite (1~63) in the ROM but has a 279K equivalent gate count due to the RAM necessary to generate Legendre sequences on the fly and store the generated Legendre sequences, and due to the generation circuit. Next, the Weil-generated universal code generator [17] has a 46K equivalent gate count because the RAM was completely removed, although it uses a ROM larger than that of the Legendre-generated universal code generator [16].

Table 5. Synthesis results.

Metrics	MB UCG [14,15]	LG UCG [16]	WG UCG [17]	Proposed UCG
Operating frequency	200 MHz	200 MHz	200 MHz	200 MHz
Critical Path Delay	4.91 ns	4.92 ns	4.93 ns	4.92 ns
Equivalent Gate Count	925K	279K	46K	19K
Normalized EGC <sup>a</sup>	100%	30.1%	4.97%	2.05%
Max Additional CC <sup>b</sup>	0	0	6924 CC	6 CC
Throughput <sup>c</sup>	200 Mbps	200 Mbps	200 Mbps	33 Mbps

<sup>a</sup> EGC: Equivalent Gate Counts; <sup>b</sup> CC: Clock Cycles; <sup>c</sup> Throughput: no degradation when throughput is higher than 1.024 Mbps.

Lastly, the proposed universal code generator reduced the hardware complexity to a 19K equivalent gate count by sharing the Legendre ROMs and associated circuitry that is used repeatedly in the Weil-generated universal code generator [17]. In summary, the proposed universal code generator has an area reduced by 98%, 93%, and 60% compared to the memory-based universal code generator [14,15], the Legendre-generated universal code generator [16], and the Weil-generated universal code generator [17], respectively. From the viewpoint of the system, if the code generator in the GNSS receiver provides throughput not smaller than several Mbps, the entire system is not degraded. The proposed universal code generator requires six clock cycles per code bit at the maximum for time

multiplexing, but there is no degradation in the overall receiver performance. The code rate of GPS L1C signals and BDS B1C signals is 1.023 Mbps, and the proposed universal code generator has a code rate of 33 Mbps. Consequently, the proposed code generator has the least hardware complexity by removing redundant hardware without affecting the overall system performance.

## 6. Conclusions

This paper proposed an area-efficient universal code generator for GPS L1C signals and BDS B1C signals. Previously, MB UCG [14,15] stored all the PRN codes by signal, channel, and satellite in the ROM, leading to the largest hardware complexity. To mitigate the huge hardware complexity, LG UCG [16] generates the Legendre sequences on the fly and stores the generated Legendre sequences in the RAM. Based on the stored Legendre sequences in the RAM, the PRN codes are generated. Compared to MB UCG [14,15], LG UCG [16] saves hardware by generating the Legendre sequence on the fly. However, LG UCG [16] demands significant hardware complexity due to the use of RAM. Recently, WG UCG [17] has generated PRN codes with Legendre sequences of ROM instead of RAM. By utilizing the characteristics of ROM, the hardware complexity and initialization time for the RAM are reduced. Finally, the main idea of the proposed universal code generator is to apply a time-multiplexing technique to the previous universal code generator in order to save the common hardware resources. The proposed structure shares the common hardware according to the scheduling of a time unit instead of using the same hardware at the same time by duplicating the same hardware. Although time multiplexing can lower the hardware complexity at the cost of increasing the latency, the proposed code generator is carefully designed in order not to degrade the system performance. As a result of the synthesis using the CMOS 65 nm process, the proposed code generator has an area reduced by 98%, 93%, and 60% compared to the memory-based universal code generator [14,15], the Legendre-generation universal code generator [16], and the Weil-generation universal code generator [17], respectively. Among the current GNSS signals, as shown in Table 1, the proposed universal code generator can support only GPS L1C and BDS B1C signals since the two GNSS signals are based on the Legendre sequence. However, it can be seamlessly extended if Legendre sequence-based codes are added to the GNSS in the future regardless of the RF band.

**Author Contributions:** Conceptualization, H.Y.; methodology, J.P. and H.Y.; software, J.P.; validation, J.P. and H.Y.; formal analysis, J.P. and H.Y.; investigation, J.P., M.K. and G.J.; resources, J.P., M.K. and G.J.; data curation, J.P., M.K. and G.J.; writing—original draft preparation, J.P.; writing—review and editing, J.P. and H.Y.; visualization, J.P.; supervision, H.Y.; project administration, H.Y.; funding acquisition, H.Y. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by Navcours Co., Ltd (Daejeon, Korea).

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

**Acknowledgments:** This work was supported by Navcours Co., Ltd. and National R&D Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science and ICT (2020M3H2A1078119).

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Manosas-Caballu, M.; Seco-Granados, G. Alternative Implementations of the GNSS Power-Based Capon Beamformer. *IEEE Signal Process. Lett.* **2021**, *28*, 1435–1439. [[CrossRef](#)]
2. Carreno-Luengo, H.; Camps, A.; Ruf, C.; Floury, N.; Martin-Neira, M.; Wang, T.; Khalsa, S.J.; Clarizia, M.P.; Reynolds, J.; Johnson, J.; et al. The IEEE-SA Working Group on Spaceborne GNSS-R: Scene Study. *IEEE Access* **2021**, *9*, 89906–89933. [[CrossRef](#)]

3. Tsui, J.B.-Y. *Fundamentals of Global Positioning System Receivers: A Software Approach*, 2nd ed.; John Wiley & Sons: New York, NY, USA, 2005; Volume 173.
4. Interface Specification IS-GPS-800H. NAVSTAR GPS Space Segment/User Segment L1C Interfaces. Available online: <https://www.gps.gov/technical/icwg/> (accessed on 28 July 2021).
5. Galileo Open Service Signal-In-Space Interface Control Document. Available online: <https://www.gsc-europa.eu/electronic-library/programme-reference-documents> (accessed on 28 July 2021).
6. Global Navigations Satellite System GLONASS. Navigational Radio Signal in Bands L1,L2. Available online: [http://russianspacesystems.ru/wp-content/uploads/2016/08/ICD\\_GLONASS\\_eng\\_v5.1.pdf](http://russianspacesystems.ru/wp-content/uploads/2016/08/ICD_GLONASS_eng_v5.1.pdf) (accessed on 28 July 2021).
7. BeiDou Navigation Satellite System Signal In Space Interface Control Document Open Service Signal. Available online: <http://en.beidou.gov.cn/SYSTEMS/ICD/> (accessed on 28 July 2021).
8. Cao, K.; Wang, L.; Li, B.; Ma, H. A Real-Time Phase Center Variation Compensation Algorithm for the Anti-Jamming GNSS Antennas. *IEEE Access* **2020**, *8*, 128705–128715. [[CrossRef](#)]
9. Ceccato, M.; Formaggio, F.; Laurenti, N.; Tomasin, S. Generalized Likelihood Ratio Test for GNSS Spoofing Detection in Devices With IMU. *IEEE Trans. Inf. Forensics Secur.* **2021**, *16*, 3496–3509. [[CrossRef](#)]
10. Bu, J.; Yu, K.; Qian, N.; Zuo, X.; Chang, J. Performance Assessment of Positioning Based on Multi-Frequency Multi-GNSS Observations: Signal Quality, PPP and Baseline Solution. *IEEE Access* **2020**, *9*, 5845–5861. [[CrossRef](#)]
11. Hussain, A.; Ahmed, A.; Magsi, H.; Tiwari, R. Adaptive GNSS Receiver Design for Highly Dynamic Multipath Environments. *IEEE Access* **2020**, *8*, 172481–172497. [[CrossRef](#)]
12. Fan, P.; Cui, X.; Zhao, S.; Liu, G.; Lu, M. A Two-Step Stochastic Hybrid Estimation for GNSS Carrier Phase Tracking in Urban Environments. *IEEE Trans. Instrum. Meas.* **2021**, *70*, 1–18. [[CrossRef](#)]
13. Mikhaylova, O.K.; Korogodin, I.V.; Lipa, I.V. Universal ranging code generator of GNSS signals. In Proceedings of the 2020 International Youth Conference on Radio Electronics, Electrical and Power Engineering (REEPE), Moscow, Russia, 12–14 March 2020.
14. Shivaramaiah, N.C.; Andrew, G.D. On the baseband hardware complexity of modernized GNSS receivers. In Proceedings of the Proceedings of 2010 IEEE International Symposium on Circuits and Systems, Paris, France, 30 May–2 June 2010.
15. Fortin, M.-A.; Landry, R., Jr. Implementation strategies for a universal acquisition and tracking channel applied to real GNSS signals. *Sensors* **2016**, *16*, 624. [[CrossRef](#)] [[PubMed](#)]
16. Yang, Y.; Ba, X.; Chen, J. A novel VLSI architecture for multi-constellation and multi-frequency GNSS acquisition engine. *IEEE Access* **2018**, *7*, 655–665. [[CrossRef](#)]
17. Park, J.; Jo, G.; Kim, J.; Yoo, H. Area-Efficient On-the-Fly Code Generator for BDS B1C Receivers. In Proceedings of the 2021 International Conference on Electronics, Information, and Communication (ICEIC), Jeju, Korea, 31 January–3 February 2021.