Modular Arithmetic CPM for SDR Platforms

Abisaí Ramírez-Pérez, *Member, IEEE*, Rodrigo Aldana-López, *Graduate Student Member, IEEE*,
Omar Longoria-Gandara, *Member, IEEE*, Jose Valencia-Velasco, *Member, IEEE*,
Luis Pizano-Escalante, *Member, IEEE*, and Ramon Parra-Michel, *Member, IEEE*

***Abstract*—This brief presents a new Continuous Phase Modulation (CPM) module for multi-*h* transmission based on modular arithmetic processing. It is well known that CPM systems are sensitive to changes in the modulation index, which is implied when fixed-point format is used in the hardware implementation. In this scenario, modular arithmetic allows a more accurate signal processing regarding SQNR performance. This proposal has been shown to improve performance compared to other state-of-the-art methods, even when the modulation indices do not have an exact representation in a given fixed-point format. Due to its highly reconfigurable architecture, this hardware block is suitable for Software Defined Radio (SDR) platforms. A complete digital architecture is also provided. Implementation results show that the new proposal have clear advantages in terms of hardware resources and operating frequency against other CPM transmitters in open-literature.**

***Index Terms*—Continuous phase modulation, multi-h CPM, reconfigurable transmitter architecture, software-defined Radio.**

I. Introduction

Continuous phase modulation (CPM) is a digital modulation format that keeps a constant envelope while presenting high spectral efficiency by smoothing the phase transitions [1]. The interest and efficiency of CPM techniques has been confirmed by industrial communications standards. For example, Bluetooth uses GFSK [2], MILSTD-188-181B for voice-over-satellite communications [3] uses *dual-h* CPM and IRIG-106-15 for aeronautical telemetry [4] employs SOQPSK-TG. More recently, CPM has emerged as an alternative modulation scheme for resolving problems in GPS codes [5], [6] and for future satellite broadcasting [7]–[9].

Furthermore, a popular practical implementation computes the phase signal offline and stores it in memory, achieving a high precision CPM signal shaping at the expense of an increment in hardware memory resources required [10], [11]. In contrast, in [12], [13] a direct hardware implementation of the CPM signal is proposed, reducing the required memory but increasing the DSP operations. More recently, [14] proposes a transmitter with a better trade-off between memory and DSP operations. In addition, this scheme is highly reconfigurable, where the CPM signal shaping and the modulation index can be modified by software, making it an appealing alternative for Software Defined Radio (SDR). However, the SQNR performance of this transmitter is degraded when the modulation indices do not have an exact representation in a given fixed-point format, e.g., if $h = 1/3$. This quantization noise is boosted when computing the CPM signal cumulated phase.

Motivated by this discussion, and with the purpose of maintain a good trade-off between memory and DSP operations while achieving high SQNR performance for the CPM signaling, a new CPM transmitter is presented. The novelty of this scheme is that modular arithmetic [15, Ch. 11] units are used to optimize the DSP operations and to handle different rational modulation indices without losing precision. This proposal provides a new reconfigurable transmitter for multi-*h* CPM systems that do not have performance degradation regarding SQNR as a figure of merit and require fewer hardware resources than other state-of-the-art CPM transmitters.

In summary, two main contributions are provided in order to achieve the goal. The first contribution consists of representing the cummulated phase term of the CPM signal as a finite set of points in a constellation diagram of a digital modulation scheme. The advantage of this approach with respect to the works [12]–[14] is that the points can be stored instead of being calculated. Moreover, the amount of memory required is drastically reduced when compared to the memory based approach in [10], [11]. The second contribution is a detailed description of the new CPM transmitter, hereafter called MA-DSP, which is based on Modular Arithmetic (MA) Digital Signal Processing (DSP) and memory elements. In addition, MA-DSP is provisioned with high reconfigurability capabilities; thus, it is a suitable and potential candidate to explore in SDR applications like the ones described in [16]–[18].

The remaining of this brief is organized as follows. Section II describes the CPM signal model. Section III presents the two contributions. A comparison of the required hardware resources between the state-of-the-art transmitters and the MA-DSP transmitter is presented in Section IV. Furthermore, SQNR performance is analyzed in Section V. This brief concludes with some final remarks in Section VI.

Manuscript received December 15, 2021; accepted January 10, 2022. Date of publication January 18, 2022; date of current version March 28, 2022. This brief was recommended by Associate Editor H.-G. Yang. *(Corresponding author: Rodrigo Aldana-López.)*

Abisaí Ramírez-Pérez, Omar Longoria-Gandara, and Luis Pizano-Escalante are with the Department of Electronics, Systems and IT, ITESO, Tlaquepaque 45604, Mexico (e-mail: abisai@iteso.mx; olongoria@iteso.mx; luispizano@iteso.mx).

Rodrigo Aldana-López is with the DIIS, UNIZAR, 50009 Zaragoza, Spain (e-mail: 794214@unizar.es).

Jose Valencia-Velasco is with the Department of Electronics, CETI, Guadalajara 44638, Mexico (e-mail: josev@ceti.mx).

Ramon Parra-Michel is with the Department of Telecommunications, CINVESTAV, Zapopan 45019, Mexico (e-mail: rparra@gdl.cinvestav.mx).

Digital Object Identifier 10.1109/TCSII.2022.3143762

II. CPM Signal Model

Following from [19, Ch. 2.2], consider an input stream of M-ary symbols $\alpha_n \in \{\pm 1, \pm 3, \ldots, \pm M-1\}$ for $n = 0, 1, \ldots$ represented in compact form by $\boldsymbol{\alpha} = \{\alpha_0, \alpha_1, \ldots\}$. The CPM model takes the symbols $\boldsymbol{\alpha}$ to produce a signal $s(t;\boldsymbol{\alpha}) := \cos(2\pi f_c t + \theta_0 + \theta(t;\boldsymbol{\alpha}))$ where f_c is the carrier frequency, θ_0 is the initial phase of the carrier and $\theta(t;\boldsymbol{\alpha})$ is the phase signal. In addition, it is advantageous to decompose $s(t;\boldsymbol{\alpha})$ as $s(t;\boldsymbol{\alpha}) \equiv \cos(2\pi f_c t + \theta_0)\mathrm{I}(t;\boldsymbol{\alpha}) - \sin(2\pi f_c t + \theta_0)\mathrm{Q}(t;\boldsymbol{\alpha})$ where $\mathrm{I}(t;\boldsymbol{\alpha}) := \cos(\theta(t;\boldsymbol{\alpha}))$ and $\mathrm{Q}(t;\boldsymbol{\alpha}) := \sin(\theta(t;\boldsymbol{\alpha}))$ are referred as the baseband components of $s(t;\boldsymbol{\alpha})$. Moreover, consider N integers $m_0, \ldots, m_{N-1} \in \mathbb{Z}$ which are used to define the general multi-h form of the phase signal given by

$$\theta(t;\boldsymbol{\alpha}) := 2\pi \sum_{n=0}^{\infty} \left(\frac{2\lambda_n}{p}\right) \alpha_n q(t - nT) \tag{1}$$

where $p \in \mathbb{Z}$ and $\lambda_n := m_{\mathrm{mod}(n,N)}$ is a cyclic sequence over $\{m_0, \ldots, m_{N-1}\}$. Furthermore, T is the symbol period, $q(t) = \int_{-\infty}^{t} g(\tau)d(\tau)$ is known as the phase smoothing response (PSR) and $g(t)$ is known as the frequency response function (FRF). Note that in the phase signal (1) there are N modulation indices $\{h_0, \ldots, h_{N-1}\} \in \mathbb{Q}$ where $h_i = 2m_i/p$, $i = 0, \ldots, N-1$, selected according to λ_n. Now, write (1) for $t \in [jT, (j+1)T)$ as

$$\hat{\theta}(t;\boldsymbol{\alpha}) := \frac{\theta(t;\boldsymbol{\alpha})}{2\pi} = \hat{\theta}_{\mathrm{ma}}(t;\boldsymbol{\alpha}) + \hat{\theta}_{\mathrm{ar}}(t;\boldsymbol{\alpha}) \tag{2}$$

where $\hat{\theta}_{\mathrm{ma}}(t;\boldsymbol{\alpha}) := \sum_{n=j-L+1}^{j} (2\lambda_n/p)\alpha_n q(t-nT)$ and $\hat{\theta}_{\mathrm{ar}}(t;\boldsymbol{\alpha}) := \sum_{n=0}^{j-L} (2\lambda_n/p)\alpha_n q(t-nT)$ are the referred as the moving-average and auto-regressive terms respectively. To simplify (2), note that the PSR $q(t)$ is a monotonically increasing function in the interval $t \in [0, LT]$ where L is the duration in symbol periods of the PSR. Moreover, $q(t) = 0$ for $t < 0$, and $q(t) = 1/2$ for $t > LT$. Therefore, if $t \in [jT, (j+1)T)$, then $q(t - nT) = 1/2$ whenever $n \leq j - L$. Henceforth, $\hat{\theta}_{\mathrm{ar}}(t;\boldsymbol{\alpha})$ in (2) can be rewritten as $\hat{\theta}_{\mathrm{ar}}(t;\boldsymbol{\alpha}) = \sum_{n=0}^{j-L} \lambda_n\alpha_n/p$. Denoting with $[\![\bullet]\!]_r := \mathrm{mod}(\bullet, r)$ for any $r > 0$, note that $\cos(2\pi\bullet) \equiv \cos(2\pi[\![\bullet]\!]_1)$ and $\sin(2\pi\bullet) \equiv \sin(2\pi[\![\bullet]\!]_1)$. In consequence, it is enough to consider $[\![\hat{\theta}(t;\boldsymbol{\alpha})]\!]_1$ instead of $\hat{\theta}(t;\boldsymbol{\alpha})$ when computing the baseband components, i.e., $\mathrm{I}(t;\boldsymbol{\alpha}) \equiv \cos(2\pi[\![\hat{\theta}(t;\boldsymbol{\alpha})]\!]_1)$ and $\mathrm{Q}(t;\boldsymbol{\alpha}) \equiv \sin(2\pi[\![\hat{\theta}(t;\boldsymbol{\alpha})]\!]_1)$. In addition, as given in [20, eq. (5)], the operator $[\![\bullet]\!]_r$ satisfies that $[\![x+y]\!]_r \equiv [\![[\![x]\!]_r + [\![y]\!]_r]\!]_r =: x \oplus_r y$ for any $x, y \in \mathbb{R}$, where the operator $\oplus_r$ is introduced to simplify notation in further developments. Hence, $[\![\hat{\theta}(t;\boldsymbol{\alpha})]\!]_1 = \hat{\theta}_{\mathrm{ma}}(t;\boldsymbol{\alpha}) \oplus_1 \hat{\theta}_{\mathrm{ar}}(t;\boldsymbol{\alpha})$.

Now, consider the discrete time version of $[\![\hat{\theta}(t;\boldsymbol{\alpha})]\!]_1$. Denote with D_s the amount of samples per symbol and define $\hat{\theta}[k;\boldsymbol{\alpha}] := \hat{\theta}(kT/D_s)$ as well as equivalent definitions for $\hat{\theta}_{\mathrm{ma}}[k;\boldsymbol{\alpha}]$ and $\hat{\theta}_{\mathrm{ar}}[k;\boldsymbol{\alpha}]$ to obtain

$$\begin{aligned} [\![\hat{\theta}[k;\boldsymbol{\alpha}]]\!]_1 &= \hat{\theta}_{\mathrm{ma}}[k;\boldsymbol{\alpha}] \ \oplus_1 \ \hat{\theta}_{\mathrm{ar}}[k;\boldsymbol{\alpha}] \\ \hat{\theta}_{\mathrm{ma}}[k;\boldsymbol{\alpha}] &= \sum_{n=j-L+1}^{j} \frac{2\lambda_n}{p}\alpha_n q[k - nD_s] \\ `\hat{\theta}_{\mathrm{ar}}[k;\boldsymbol{\alpha}] &= \sum_{n=0}^{j-L} \frac{\lambda_n\alpha_n}{p} \end{aligned} \tag{3}$$

for $jD_s \leq k < (j+1)D_s$ where $q[k] := q(kT/D_s)$.

III. MA-DSP CPM Transmitter

A. High SQNR Auto-Regresive Signal Computation

This new proposal takes the strengths of previous proposed transmitters: the high reconfigurability and low-hardware resources of the CPM transmitter in [14] and the high SQNR of the CPM transmitter in [10], hereinafter referred as DPS-based and ROM-based CPM transmitters respectively. In addition, the experiments with modulation indices that do not have an exact representation using a given fixed-point format, e.g., $h = 1/3$ or $h = 1/6$, show that SQNR performance of DSP-based [14] is degraded. This quantization noise is going to increase further when the CPM signal cumulated phase is computed. Once identified that the loss of SQNR performance of the DSP-based CPM transmitter is attributable to the auto-regressive term, an analysis of the auto-regressive term, $\hat{\theta}_{\mathrm{ar}}[k;\boldsymbol{\alpha}]$ is performed in the following. First, note that the modulo operator complies $[\![\frac{1}{p}\bullet]\!]_1 = \frac{1}{p}[\![\bullet]\!]_p$, $\forall p \in \mathbb{Z}$. Therefore, $[\![\hat{\theta}_{\mathrm{ar}}[k;\boldsymbol{\alpha}]]\!]_1 = \frac{1}{p}[\![\sum_{n=0}^{j-L} \lambda_n\alpha_n]\!]_p$.

Given that α_n and λ_n can only take M and N different integer values respectively, the set of all possible products $\alpha_n\lambda_n \in \mathbb{Z}$ is of size at most MN. Moreover, for any $v \in \mathbb{Z}$ it follows that $[\![v]\!]_p \in \{0, 1, \ldots, p-1\}$. Hence: $[\![\hat{\theta}_{\mathrm{ar}}[k;\boldsymbol{\alpha}]]\!]_1 \in \{0, \frac{1}{p}, \ldots, \frac{p-1}{p}\}$. This analysis implies that the term $[\![\hat{\theta}_{\mathrm{ar}}[k;\boldsymbol{\alpha}]]\!]_1$ can be computed offline and stored using pB_s bits, where B_s are the amount of bits per sample. In addition, the auto-regressive term can be decomposed as

$$p[\![\hat{\theta}_{\mathrm{ar}}[k;\boldsymbol{\alpha}]]\!]_1 = \left(\lambda_{j-L}\alpha_{j-L}\right) \oplus_p \sum_{n=0}^{j-L-1} \lambda_n\alpha_n \tag{4}$$

where using $\hat{\theta}_{\mathrm{ar}}[k - D_s;\boldsymbol{\alpha}]$ in (3), the last term in (4) coincides with $p\hat{\theta}_{\mathrm{ar}}[k - D_s;\boldsymbol{\alpha}] = \sum_{n=0}^{j-L-1} \lambda_n\alpha_n$ since $(j-1)D_s \leq k - D_s < jD_s$. Therefore, (4) can be written as

$$p[\![\hat{\theta}_{\mathrm{ar}}[k;\boldsymbol{\alpha}]]\!]_1 = \lambda_{j-L}\alpha_{j-L} \oplus_p p[\![\hat{\theta}_{\mathrm{ar}}[k - D_s;\boldsymbol{\alpha}]]\!]_1 \tag{5}$$

since $[\![p\hat{\theta}_{\mathrm{ar}}[k - D_s;\boldsymbol{\alpha}]]\!]_p = p[\![\hat{\theta}_{\mathrm{ar}}[k - D_s;\boldsymbol{\alpha}]]\!]_1$. Consider the following two key ideas which are important for the implementation of the MA-DSP CPM transmitter. First, evaluating (5) at $k = jD_s$ leads to

$$p[\![\hat{\theta}_{\mathrm{ar}}[jD_s;\boldsymbol{\alpha}]]\!]_1 = \lambda_{j-L}\alpha_{j-L} \oplus_p p[\![\hat{\theta}_{\mathrm{ar}}[(j-1)D_s;\boldsymbol{\alpha}]]\!]_1 \tag{6}$$

which provides only the first sample of (5) in the interval $jD_s \leq k < (j+1)D_s$. Second, note that $p[\![\hat{\theta}_{\mathrm{ar}}[k;\boldsymbol{\alpha}]]\!]_1$ remains constant in the whole interval $jD_s \leq k < (j+1)D_s$.

Henceforth, to compute (5) it is enough to recursively generate the discrete under-sampled signal $p[\![\hat{\theta}_{\mathrm{ar}}[jD_s;\boldsymbol{\alpha}]]\!]_1$ for $j \in \mathbb{Z}$, according to (6) and over-sample the result by fitting $D_s - 1$ zeros between each sample. Now, a convolution with a unit pulse $\Pi[k; D_s]$ of D_s samples defined by $\Pi[k; D_s] = 1$ for $0 \leq k < D_s$ and $\Pi[k; D_s] = 0$ otherwise, is then applied to the under-sampled signal obtaining (5).

Once the value of $p[\![\hat{\theta}_{\mathrm{ar}}[k;\boldsymbol{\alpha}]]\!]_1 \in \{0, \ldots, p-1\}$ is obtained, the result can be used to map directly to the corresponding value of $[\![\hat{\theta}_{\mathrm{ar}}[k;\boldsymbol{\alpha}]]\!]_1$ without any division operation. This is done by storing $\{0, 1/p, \ldots, (p-1)/p\}$ in a memory and using $p[\![\hat{\theta}_{\mathrm{ar}}[k;\boldsymbol{\alpha}]]\!]_1$ as the address. Hence, the value of $[\![\hat{\theta}_{\mathrm{ar}}[k;\boldsymbol{\alpha}]]\!]_1$ can be recovered with constant SQNR which increases with

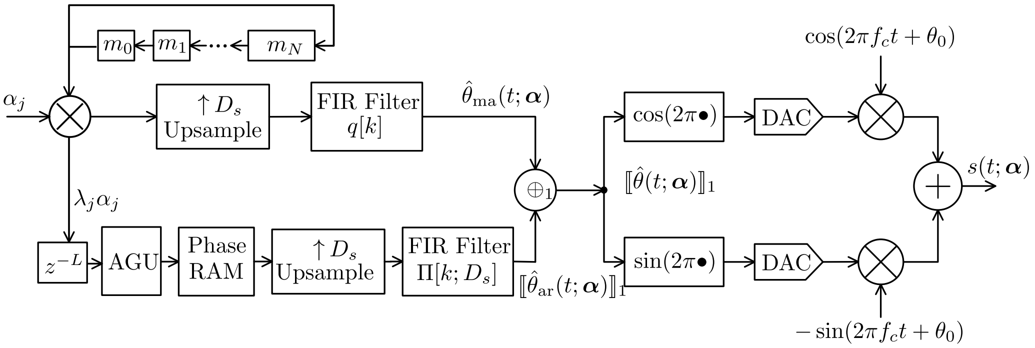

m_0
m_1
m_N
α_j
$\uparrow D_s$ Upsample
FIR Filter $q[k]$
$\hat{\theta}_{\rm ma}(t;\boldsymbol{\alpha})$
$\cos(2\pi f_c t + \theta_0)$
$\cos(2\pi\bullet)$
DAC
$s(t;\boldsymbol{\alpha})$
$\oplus_1$
$[\![\hat{\theta}(t;\boldsymbol{\alpha})]\!]_1$
$\lambda_j\alpha_j$
z^{-L}
AGU
Phase RAM
$\uparrow D_s$ Upsample
FIR Filter $\Pi[k;D_s]$
$[\![\hat{\theta}_{\rm ar}(t;\boldsymbol{\alpha})]\!]_1$
$\sin(2\pi\bullet)$
DAC
$-\sin(2\pi f_c t + \theta_0)$

Fig. 1. Proposed architecture for MA-DSP CPM transmitter.

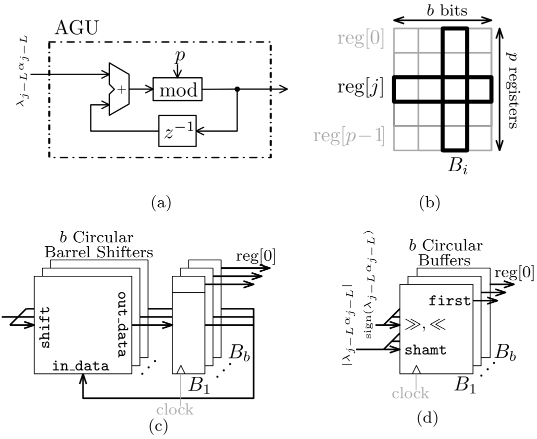
B_s and does not suffer from any additional SQNR degradation performance as opposed to the scheme presented in [14].

B. CPM Transmitter Implementation

The structure of the MA-DSP CPM transmitter is shown in Figure 1. Based on modular arithmetic processing, this new transmitter can generate single-*h* and multi-*h* CPM waveforms. As derived Section III-A, the main advantage of MA-DSP is that it does not suffer SQNR degradation performance since the values of the auto-regressive term are computed offline and stored in memory. In addition, it preserves the high reconfigurability of the DSP-based CPM transmitter. Note that the baseband components are computed through $\cos(2\pi\bullet) : [0, 1] \to [-1, 1]$ and $\sin(2\pi\bullet) : [0, 1] \to [-1, 1]$ which are implemented via piecewise polynomial approximations. The moving-average computation block and the trigonometric blocks are the same as in the DSP-based CPM transmitter presented in [14]. However, the computation of the auto-regressive term, is implemented in a different fashion.

As argued before, (6) is used to compute the address of a RAM which stores the possible values $[\![\hat{\theta}_{\rm ar}[k;\boldsymbol{\alpha}]]\!]_1$. Hence, the implementation block for (6) is called Address Generator Unit (AGU). As shown in Figure 1, $\alpha_j\lambda_j$ is delayed by L units by means of a z^{-L} block in order to obtain $\lambda_{j-L}\alpha_{j-L}$ as required in (6). Consequently, $\lambda_{j-L}\alpha_{j-L}$ is fed to the AGU module whose output is the value of the under-sampled address signal $p[\![\hat{\theta}_{\rm ar}[jD_s;\boldsymbol{\alpha}]]\!]_1$. The RAM stores $\{0, 1/p, \ldots, (p-1)/p\}$ and outputs the signal $[\![\hat{\theta}_{\rm ar}[jD_s;\boldsymbol{\alpha}]]\!]_1$ which is fed to an up-sampler implemented by a FIR filter with impulse response $\Pi[k; D_s]$.

The functional diagram of the AGU unit is shown in Figure 2-(a), which corresponds directly to (6), requiring an abstract $\mathrm{mod}(\bullet, p)$ unit. Note that the $\mathrm{mod}(\bullet, p)$ operator as well as the modular sum $\oplus_p$ can be implemented using techniques as the ones in [21], [15, Ch. 11]. However, this design can be simplified by taking advantage of the structure of the whole AGU. An alternative is to introduce an array of p registers which are initialized with the values $0, \ldots, p-1$ represented using b bits. This array of registers can be decomposed in a matrix as shown in Figure 2-(b), where each row represents the elements of the array ref[j], $j = 0, \ldots, p-1$. Moreover, the i-th column denoted with $B_i, i = 1, \ldots, b$ represents the block of p bits containing the i-th bit of all ref[1], ..., ref[$p-1$]. Thus, in order to implement the modular sum in Figure 2-(a), a stack of b circular barrel shifters is used, one for each block B_i, $i = 1, \ldots, b$. The barrel shifter unit has a `shift` port which takes a signed input. Hence, the shift amount and shift direction are computed using the magnitude and sign of `shift`. Finally, the output of the AGU is taken as reg[0]. This idea can also be realized by promoting the register blocks $B_1, \ldots, B_b$ to circular buffers as shown in Figure 2-(d), implementing the same functionality as before.



(a) (b) (c) (d)

Fig. 2. Address Generating Unit (AGU) implementation. (a) Functional diagram of the AGU. (b) Matrix representation of the register array of p elements with b bits each. (c) AGU implementation using b circular barrel shifters and register array. (d) AGU implementation using b circular buffers.

C. Detailed Architecture

A detailed architecture of the proposed MA-DSP CPM transmitter is shown in Figure 3. For configuration purposes, the control module has the `Conf_C` input signal. `Conf_C=1` sets the CPM signal parameters, whereas `Conf_C=0` enables the operation (run) mode of the modulator. In what follows, these two modes are described in detail. In configuration mode, the user feeds the CPM signal parameters D_s, N, L and p to the Control Module and the required data is stored in each of four RAMs: FIR coefficients $\{q[0], q[1], \ldots, q[LD_s-1]\}$, numerators of modulation indices $\{m_0, \ldots, m_{N-1}\}$, the modulation indices $\{h_0, \ldots, h_{N-1}\}$ and auto-regressive term possible values $\{0, 1/p, \ldots, (p-1)/p\}$. The procedure for storing samples in RAMs is the following: write enable signals (`we_F`, `we_R`, `we_K` and `we_P`) are set and the values to be stored are introduced in their respective data port (`Coeff_F`, *h*, `DataIn_K`, and `DataIn_P`), and location addresses are given in the ports (`Addr_F`, `Addr_R`, `Addr_K` and `Addr_P`). In run mode,

TABLE I
REQUIRED HARDWARE RESOURCES FOR SOQPSK-TG CPM TRANSMITTER IMPLEMENTATION

Name	**Slice Registers**	**Slice LUTs**	**LUTs as logic**	**LUTs as memory**	**Occupied Slices**	**LUT FF pairs**	**Memory (in bits)**	**DSP**	**Frequency (MHz)**	**Rate=Frequency/D_s (Mbps)**
ROM [10]	22	22	22	0	15	23	1572864	0	298.77	74.6
INT [13]	1154	1078	1046	32	346	1204	0	17	NA	NA
DSP [14]	1322	1008	846	107	469	1372	0	15	112.17	28.0
MA-DSP	1030	891	723	130	381	1114	0	17	151.65	37.91

NA: Not available

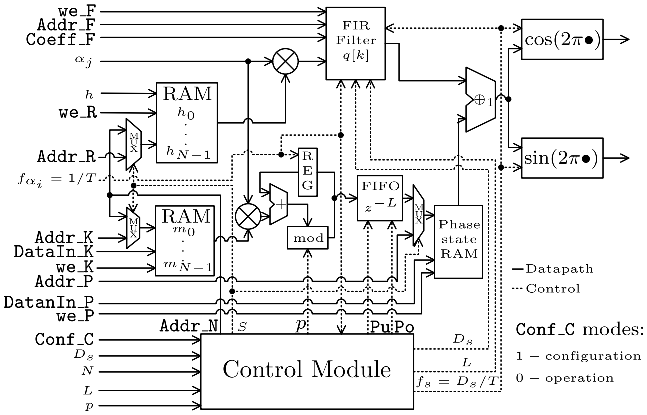


Fig. 3. Detailed hardware architecture of the MA-DSP CPM transmitter.

the incoming symbol is processed in two sections: moving-average and auto-regressive. In the first section, the incoming symbol α_j is multiplied by the selected numerator of the modulation index λ_j, which is retrieved from the RAM using the `Addr_N` generated by the Control Module. The selector `S` is cleared and the addressing signal is generated via a module-N counter. Then, $\lambda_j\alpha_j$ is convolved with the coefficients of the FIR filter internally handling two rate domains: $1/T$ and D_s/T.

In the second section, the product $\lambda_j\alpha_j$ is fed to the AGU block. The AGU is implemented using Figure 2-(a) where the z^{-1} delay block is replaced by a parallel-in parallel-out register (REG) and a reconfigurable modulo-p operation block (mod) is used. The result of the modular sum is stored in a FIFO memory that generates a delay of L symbol intervals using push `Pu` and pop `Po` signals. The delayed value addresses the phase state RAM delivering the corresponding auto-regressive term. Both FIR filter and phase state RAM outputs are added in a fractional adder in order to obtain $[\![\hat{\theta}[k;\boldsymbol{\alpha}]]\!]_1$. The result is passed to trigonometric blocks, computed with piecewise polynomial approximations as in [14].

IV. HARDWARE SYNTHESIS AND MEASUREMENTS

Table I reports hardware resources used in the implementation of four transmitters. The approaches are the integration-based (INT-based) as in [13] that uses a direct implementation of the CPM signaling; ROM-based from [10] where the phase signal is computed offline and stored in memory; DSP-based approach proposed in [14] that employs digital signal processing instead of memory; and finally, the proposed MA-DSP scheme in this brief. The SOQPSK-TG standard defined in [4] was chosen for validation. The implementations have general parameters: $D_s = 4$, $B_s = 12$, $M = 4$, $L = 8$ according to the SOQPSK-TG standard and a single modulation index $1/2$, consequently $p = 4$ and $m_0 = 1$. In addition, transmitters were coded using Verilog-HDL and synthesized in a xc7z020-2clg484 Field Programmable Gate Array (FPGA) from Xilinx vendor using the software tool ISE Design Suite 14.7. In addition, the SQNR performance metric was used to evaluate the performance of the transmitters with different modulation indices and datawidth for the system. Firstly, Figure 4 considers a modulation index of $h = 1/2$ because it can be represented exactly in fixed-point format. In contrast, Figure 5 shows the results with modulation indices $h = 1/5$ and $h = 2/3$, which cannot be represented accurately in fixed-point format. In what follows, the comparisons of synthesis and SQNR results are discussed jointly.

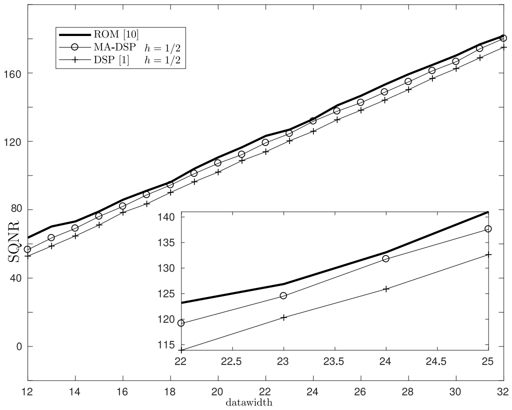


Fig. 4. SQNR performance measure for CPM transmission using $h = 1/2$ from 12 to 32 bits of datawidth fixed-point format.

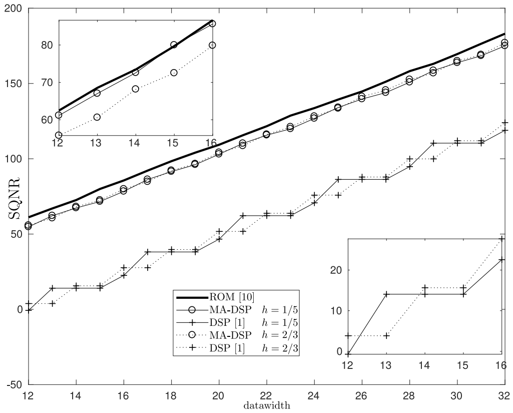


Fig. 5. SQNR performance measure for CPM transmission using $h = 1/5$ and $h = 2/3$ from 12 to 32 bits of datawidth fixed-point format.

V. COMPARISON RESULTS AND DISCUSSION

With reference to Table I, it can be observed that the ROM-based transmitter has the best performance in terms of logic elements of the FPGA, represented in LUTs. However, since this transmitter uses the whole representation of the CPM signal, it requires the largest amount of memory among the transmitters. Likewise, the accurate representation of the CPM signal allows the ROM-based approach to achieve the best SQNR performance regardless of the modulation index as shown in Figures 4 and 5. For this reason, this scheme is used as a reference model at the expense of its high memory usage, which may not desirable for some applications.

In the case of MA-DSP, and considering as a figure of merit the sum of LUT's as logic, memory, and FF pairs, Table I achieves a reduction of 15% and 14% with respect to the INT-based and DSP-based approaches, respectively. Moreover, this scheme requires a small amount of LUTs as memory to store the modulation indices and filter coefficients as shown in Figure 3. This also has an impact on the bit-rate, where the ROM-based approach achieves the highest performance, followed by MA-DSP due to the trade-off between memory and DSP operations. In this regard, MA-DSP is more economical than ROM-based in terms of memory resources. Figure 4 shows that the SQNR performance of both MA-DSP and DSP-based CPM transmitters match the one of the reference as expected, since the modulation index $h = 1/2$ can be represented exactly in fixed-point format. It can be observed that the SQNR increases linearly with the datawidth since quantization noise is reduced as datawidth increases. Quantitatively, there is an increment of 6dB per bit, which is expected from the well-known relation between SQNR and datawidth. On the other hand, the impact of an inaccurate representation of the modulation index is shown in Figure 5, where MA-DSP matches the performance of the reference scheme, even with these modulation indices. In contrast, the SQNR performance of the DSP-based transmitter is 50 dB below both MA-DSP and reference approaches. Thus, there is a trade-off between memory, DSP operations, and quality measured by SQNR that must be taken into account when choosing a CPM transmitter according to the applications. In this context, MA-DSP may be more suitable than the other approaches in applications where reconfigurability and SQNR quality are important features to take into account, e.g., in SOQPSK and GFSK, when using modulation indices that cannot be represented exactly in fixed-point format.

VI. Conclusion and Future Work

In this brief, the MA-DSP CPM transmitter was presented which combines the benefits of another state of the art approaches as well as using modular arithmetic processing for accurate signal computation. In addition, simulation results showed that MA-DSP is able to tackle the SQNR performance loss associated prior art. Moreover, the hardware architecture showed to have clear advantages against other methods in terms of hardware resources and operating frequency. Note that this modular arithmetic technique can be extended to specific CPM transmitters and receivers that are very attractive for SDR platforms as the ones in [22], [23].

References

[1] F. Xiong, *Digital Modulation Techniques*. Norwood, MA, USA: Artech House, Jan. 2006.

[2] *The Bluetooth Core Specification, V4.2*, Bluetooth Special Interest Group, Kirkland, WA, USA, 2014.

[3] B. W. Peterson, D. R. Stephens, and W. H. Tranter, "DFSE equalization of dual-h CPM over UHF MILSATCOM channels," in *Proc. Military Commun. Conf. (MILCOM)*, vol. 2. Anaheim, CA, USA, Oct. 2002, pp. 1406–1411.

[4] *Telemetry Standards*, IRIG Standard 106-15, 2015.

[5] P. Kossin, J. Zygmaniak, and T. Guasconi, "Continuous phase modulation for GPS codes: Reducing out-of-band emissions and improving RF/DC efficiency," Space Intell. Syst., Harris Corp., Melbourne, FL, USA, Rep., Jul. 2018. [Online]. Available: https://gpsworld.com/wp-content/uploads/2018/09/Harris-Corporation_-Continuous-Phase-Modulation_White-Paper-07-25-2018.pdf

[6] R. Xue, Y. Sun, and D. Zhao, "CPM signals for satellite navigation in the S and C bands," *Sensors*, vol. 15, no. 6, pp. 13184–13200, 2015. [Online]. Available: https://www.mdpi.com/1424-8220/15/6/13184

[7] Y. Zhang, Q. Li, L. Huang, K. Dai, and J. Song, "Optimal design of cascade LDPC-CPM system based on bionic swarm optimization algorithm," *IEEE Trans. Broadcast.*, vol. 64, no. 3, pp. 762–770, Sep. 2018.

[8] J. Qi, S. B. Makarov, M. Liu, B. Li, and W. Xue, "Research on an optimization method for a partially responsive continuous phase modulated (CPM) signal based on an optimal generic function," *Symmetry*, vol. 11, no. 9, p. 1114, 2019. [Online]. Available: https://www.mdpi.com/2073-8994/11/9/1114

[9] Y. Sun, "Optimal parameter design of continuous phase modulation for future GNSS signals," *IEEE Access*, vol. 9, pp. 58487–58502, 2021.

[10] J. B. Anderson, T. Aulin, and C.-E. Sundberg, *Digital Phase Modulation*, 1st ed. Boston, MA, USA: Springer, 1986.

[11] A. Linz and A. Hendrickson, "Efficient implementation of an I-Q GMSK modulator," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 43, no. 1, pp. 14–23, Jan. 1996.

[12] T. A. D. Riley and M. A. Copeland, "A simplified continuous phase modulator technique," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 41, no. 5, pp. 321–328, May 1994.

[13] D. Rieth, C. Heller, and G. Ascheid, "FPGA implementation of shaped offset QPSK modulator," in *Proc. IEEE Int. Conf. Digit. Signal Process. (DSP)*, 2015, pp. 790–793.

[14] A. Ramírez-Pérez, R. Parra-Michel, A. Rodríguez-García, and L. F. González-Pérez, "A new single h and multi-h CPM transmitter," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 63, no. 5, pp. 716–726, May 2016.

[15] R. Drechsler and M. Soeken, *Advanced Boolean Techniques: Selected Papers From the 13th International Workshop on Boolean Problems*, 1st ed. Cham, Switzerland: Springer, 2019.

[16] K. Morioka *et al.*, "An implementation of CPFSK-OFDM systems by using software defined radio," in *Proc. WAMICON*, 2014, pp. 1–3.

[17] F. harris, R. Bell, and U. Jha, "An optimum receiver for MIL-STD-188-181C CPM shaped BPSK," in *Proc. 22nd Int. Symp. Wireless Personal Multimedia Commun. (WPMC)*, 2019, pp. 1–5.

[18] A. Zaheer, M. Zeeshan, and M. Waqas, "Multiband multi-rate narrowband software defined radio waveform based on CPM," in *Proc. 22nd Int. Conf. Adv. Commun. Technol. (ICACT)*, 2020, pp. 212–216.

[19] J. Anderson, T. Aulin, and C. Sundberg, *Digital Phase Modulation* (Applications of Communications Theory). Boston, MA, USA: Springer U.S., 1986. [Online]. Available: https://books.google.es/books?id=5IjhvwEACAAJ

[20] H. L. Garner, "The residue number system," *IRE Trans. Electron. Comput.*, vol. EC-8, no. 2, pp. 140–147, 1959.

[21] J. T. Butler and T. Sasao, "Fast hardware computation of x mod z," in *Proc. IEEE Int. Symp. Parallel Distrib. Process. Workshops Ph.D. Forum*, 2011, pp. 294–297.

[22] R. Aldana-Lopez, J. Valencia-Velasco, O. Longoria-Gandara, and L. Pizano-Escalante, "Digital linear GFSK demodulator for IoT devices," *IET Commun.*, vol. 12, no. 16, pp. 1997–2004, 2018.

[23] J. Valencia-Velasco, O. Longoria-Gandara, R. Aldana-Lopez, and L. Pizano-Escalante, "Low-complexity maximum-likelihood detector for IoT BLE devices," *IEEE Internet Things J.*, vol. 7, no. 6, pp. 4737–4745, Jun. 2020.