



POWER OPTIMISATION FOR ADAPTIVE EMBEDDED WIDEBAND RADIOS

A Thesis submitted to
the School of Computer Engineering
of Nanyang Technological University

by

Pham Hung Thinh

in fulfillment of the requirement for
the Degree of Doctor of Philosophy of
for Computer Engineering

February 10, 2015

List of Figures

| | | |
|------|--|----|
| 5.1 | Baseband processing block diagram. | 6 |
| 5.2 | Pilots in the long preamble of IEEE 802.16-2009. | 9 |
| 5.3 | Circuit of known pilots shift register | 12 |
| 5.4 | Resource sharing approach for computing $V_{\hat{\epsilon}}$ | 13 |
| 5.5 | Architecture of proposed IFO estimator. | 14 |
| 5.6 | Fail rate of IFO estimation methods in AWGN channel without RTO. | 17 |
| 5.7 | Fail rate of IFO estimation methods in AWGN channel with RTO. | 18 |
| 5.8 | Fail rate of IFO estimation methods in SUI1 channel. | 18 |
| 5.9 | Fail rate of IFO estimation methods in SUI2 channel. | 19 |
| 5.10 | Fail rate for different wordlengths in AWGN channel without RTO. | 20 |
| 5.11 | Fail rate for different wordlengths in AWGN channel with RTO. | 21 |
| 5.12 | Fail rate for different wordlengths in SUI1 channel. | 21 |
| 5.13 | Fail rate for different wordlengths in SUI2 channel. | 22 |
| 5.14 | DSP block based 3-input adder for correlation. | 24 |

List of Tables

| | | |
|-----|---|----|
| 5.1 | Resource utilisation and dynamic power consumption of IFO estimators. | 24 |
|-----|---|----|

Chapter 1

Research Introduction

Chapter 2

Background Literature

Chapter 3

Multiplierless Correlator Design for low-power systems

Chapter 4

A Method for OFDM Timing Synchronisation

Chapter 5

A CFO Estimation Method for OFDM Synchronisation

5.1 Introduction and Related Work on OFDM synchronisation

The previous chapter, the robust OFDM synchronisation against a large CFO was presented. Although the timing synchronisation is able to perform well in the case of large CFO, the CFO estimation of this method only estimates the fractional CFO. The large CFO can occur as a result of the Doppler Effect and/or due to local oscillator instability. The CFO is normalised by subcarrier spacing and usually divided into an integer part (IFO), as a multiple of subcarrier spacings, and a fractional part (FFO). IFO causes a circular shift of the subcarrier in the frequency domain while FFO results in ICI because of lost orthogonality between subcarriers. The Fig. 5.1 illustrates the FFO and IFO estimation in the baseband processing of a typical OFDM system. The IFO estimation is performed after the FFT and relies upon

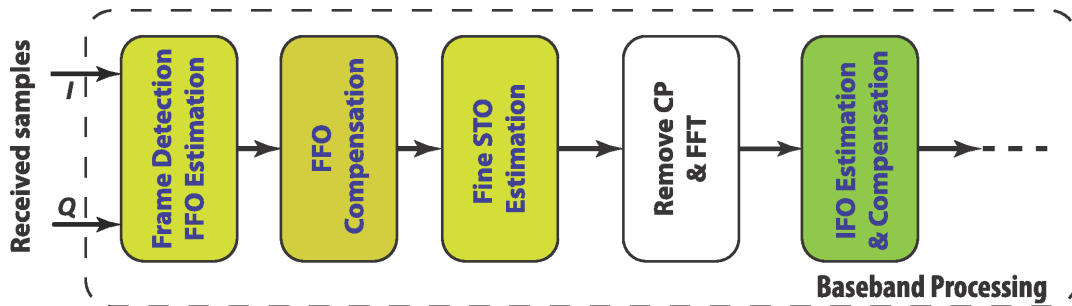


Figure 5.1: Baseband processing block diagram.

cross-correlation, which consumes significant hardware resources. A typical OFDM receiver avoids the need for IFO estimation by limiting CFO tolerance to be smaller than the range of FFO estimation, resulting in a set of very strict constraints on the design of the RF front-end. However, if CFO exceeds the range of FFO estimation, IFO is present and the system cannot function correctly. This is more pronounced in applications that subject the front-end to intensive Doppler effects, for very high carrier frequencies, or when supporting multiple frequency bands for different standards.

As mentioned in the previous chapter, the estimation range of fractional CFO has a limitation that is shown in (??); for instance its range is up to ± 2 sub-carrier spacings in the case of the IEEE 802.16 preamble. The value of the IFO is a multiple of the coarse CFO's range. It is commonly estimated by performing cross-correlation [3, 4] in the frequency domain.

Considering that the signal is transmitted over a frequency selective channel that has channel impulse response (CIR) h with length, L ($L < N_{CP}$). and it is also corrupted by AWGN. The received signal, with frequency offset and timing offset, can be expressed in the time domain as

$$y[n] = \sum_{l=0}^{L-1} h[l]x[n - \tau - l]e^{i(2\pi\xi\frac{n-\tau}{N} + \phi_0)} + w[n] \quad (5.1)$$

where $w[n]$ denotes AWGN in the time domain, τ , ϕ_0 are residual timing offset (RTO) and error phase, respectively, and ξ is the normalised CFO that can be divided in to a FFO part λ and IFO part ϵ as $\xi = \lambda + \epsilon$.

Assuming FFO and STO be compensated by earlier stages of synchronisation, as has been investigated in detail by other authors [4, 5]. The received preamble symbol after CP removal at FFT output is

$$Y[k] = e^{i(\phi_0 - 2\pi\frac{\tau k}{N})} H[k - \epsilon] X[k - \epsilon] + W[k] \quad (5.2)$$

where $W[k]$ and $H[k]$ are the frequency domain representations of AWGN and CIR, respectively.

As mentioned previously, IFO results in a cyclic shift in the frequency domain. It is commonly estimated by performing cross-correlation in the frequency domain. By contrast, RTO causes a linear phase rotation on samples in the frequency domain that may cause degradation of cross-correlation performing IFO estimation. Based on a differential demodulation of

the FFT output, the IFO can be enhancedly determined with robustness to frequency selective channel and RTO using the correlation function [6] expressed by:

$$\hat{\epsilon} = \underset{\tilde{\epsilon}}{\operatorname{argmax}} \left| \sum_{k=1}^N Y^*[k-1]Y[k]X^*[k-\tilde{\epsilon}]X[k-1-\tilde{\epsilon}] \right| \quad (5.3)$$

where $(.)^*$ denotes complex conjugation, $\hat{\epsilon}$, $\tilde{\epsilon}$ are estimated and trial values of ϵ , respectively, $Y[k]$ and $X[k]$ denote the k^{th} frequency symbol index of the received symbol and the known transmitted preamble, respectively, and the symbol size N is equal to the FFT size.

The estimated IFO can achieve high precision using cross-correlation in the frequency domain, however implementing cross-correlation clearly involves a significant hardware overhead, with a multiplier needed for each element in the cross-correlation. Sign-bit cross-correlation [7] is a widely adopted approach to reducing correlation complexity using only the most significant bit (MSB) of signed numbers in the correlation computation. In this way, complexity is reduced at the cost of performance degradation. Despite the adoption of such methods, cross-correlation remains computationally expensive, especially when dealing with a large FFT size. It should be noted here that several IFO estimation methods have been published which claim robustness to frequency selective channels and RTO. However published FPGA implementations of these methods are lacking to date, possibly because the hardware costs are considerable – even when adopting a sign-bit cross-correlation approach.

There are few practical implementations for IFO estimation and correction. A notable exception was presented in [4], in which time-domain cross correlation is performed between received samples and pre-rotated versions of the preamble corresponding possible IFO values. Although the method uses efficient sign-bit correlation to reduce hardware cost (at the cost of decreased estimation accuracy and increased sensitive to frequency selective channels), it is not efficient when the possible IFO range is large since it effectively performs an exhaustive search.

The state of the art OFDM synchronisation methods typically has a small tolerance of CFO that requires the hardware be strictly constrained to have a small CFO that doesn't introduce IFO. This requirement leads to an increase in total system cost. Particularly, in the case of MSCR such a need for precision coupled with a requirement for a wide ranging frequency band would make the RF front-end design difficult, if not impossible. In the following sections,

novel methods for timing synchronisation and IFO estimation are presented to overcome this challenge targeting to efficient and low-power systems.

5.2 Enhanced OFDM Synchronization Through Novel IFO Estimation Architecture

In this section, we present a novel method in which hardware resources and power consumption are reduced by determining IFO estimates for only a subset of possible IFO values. This in turn enables an efficient resource sharing folded architecture to be adopted. Adjusting the precision of individual correlation computations within this novel architecture leads to a fine degree of control on the trade-off between performance and power consumption. The implementation uses the long preamble of IEEE 802.16-2009 [8] for estimating IFO values. This has a symbol size of 256 with 100 pilots, as illustrated in Fig. 5.2. These 100 pilots are distributed, 50 pilots per side, at even sub-carrier spacings from 2 to 100 and from 156 to 254. The remaining sub-carriers are null. In OFDM-based systems such as 802.11, 802.16, and 802.22, the short preamble is used to estimate and compensate for FFO. Typically, fractional CFO estimation has a limited range of up to ± 2 sub-carrier spacings depending on the format of the short preamble. This results in the possible IFO values being a multiple of 4. Readers should refer to [4] to understand in detail the limitation of fractional CFO and possible IFO values. It should be noted that although the proposed method is applied to 802.16 for evaluating the accuracy of IFO estimation, this method can be applied in other OFDM-based systems having similar preamble structures, such as 802.11 and 802.22.

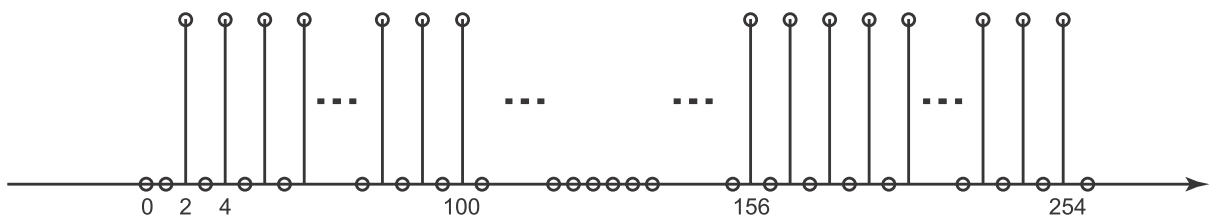


Figure 5.2: Pilots in the long preamble of IEEE 802.16-2009.

5.2.1 Proposed Algorithm

Firstly, a subset of possible IFO values is determined. We assume that the RF front end can provide CFO stability in a range from -14 to +18 subcarrier spacings, which is relaxed compared to the strict RF front end constraints in 802.16 that would typically otherwise lead to increased RF hardware costs. Generally, the larger the range of IFO estimation that is performed, the larger the CFO that the baseband system can tolerate. A wider CFO tolerance in turn leads to a relaxation in RF front-end specification, reducing system cost. In this system, there are 8 possible values for IFO estimation in the assumed CFO range. The subset of possible IFO values is denoted $S_{IFO} = \{-12, -8, -4, 0, 4, 8, 12, 16\}$. Moreover, samples are pre-offset by 12 subcarrier spacings prior to calculation to ensure that all possible values of IFO are positive, $S'_{IFO} = \{0 : 4 : 28\}$. This means that received symbols will only ever need to be shifted right to compensate IFO, thus reducing buffer memory requirements.

Secondly, a resource sharing folded architecture is designed to significantly reduce the hardware cost. Conventionally, to obtain high accuracy, IFO estimation is computed across all pilots in the preamble. This results in considerable hardware overhead, especially with a large number of pilots, as is the case for IEEE 802.16-2009. Evidently, as the number of pilots increases, the correlation result shows greater robustness to noise. However, we will demonstrate in Section 5.2.3 that the beneficial effect of calculating across additional pilots leads to decreasing gains in performance. In fact the performance improvement reaches a plateau. We therefore propose making use of only a subset of pilots, while maintaining estimation accuracy as much as possible. Then, by spreading the chosen subset of pilots carefully in the time domain, it becomes possible to share resources when computing the cross-correlation, hence reducing area and power consumption. When the proposed method is applied to IEEE 802.16-2009 offset estimation, the pilots used for the IFO computation are selected at subcarrier indices that are multiples of 4, leading to a natural four-fold resource sharing architecture. Hence, the IFO estimation can be expressed as:

$$\hat{\epsilon} = \underset{\tilde{\epsilon} \in S'_{IFO}}{\operatorname{argmax}} \left| \sum_{k=1}^{N/4} P(4k)A(4k - \tilde{\epsilon}) \right| \quad (5.4)$$

where $P(4k) = Y^*(4k - 2)Y(4k)$ denotes the correlation of two consecutive received pilots, and $A(4k) = X(4k - 2)X^*(4k)$ is the correlation of two consecutive transmitted pilots that can be pre-computed.

Thirdly, although sign-bit cross-correlation is often used in conventional implementations [4, 7], as it can significantly reduce computational complexity, it also leads to reduced precision and hence reduced estimation performance, especially in the case of frequency selective channels. For this reason, we instead apply multiplierless correlation to enhance the accuracy of estimation compared to the sign-bit approach. In [9], the authors demonstrated a trade-off between cost and accuracy for multiplierless OFDM synchronisation. We further investigate this effect in the current chapter in terms of the trade-off between cost and accuracy when reducing the wordlength used to represent $P(4k)$.

5.2.2 Proposed Architecture

The proposed method divides IFO estimation into multiple repeated computations with resource sharing based upon the four-sample timing between selected spread pilots. The estimation of IFO in (5.3) can now be rewritten as follows:

$$\begin{aligned}\hat{\epsilon} &= \underset{\tilde{\epsilon} \in S'_{IFO}}{\operatorname{argmax}} |V_{\tilde{\epsilon}}|, \\ V_{\tilde{\epsilon}} &= \sum_{k=0}^{25} P(4k)A_{\tilde{\epsilon}}(4k) + P(L + 4k)A_{\tilde{\epsilon}}(L + 4k),\end{aligned}\tag{5.5}$$

where $A_{\tilde{\epsilon}}$ denotes the correlation of two consecutive pre-rotated known pilots corresponding to one IFO value, and $V_{\tilde{\epsilon}}$ is the cross-correlation between received pilots and pre-rotated known pilots. Since the pilots of the long preamble are distributed on two sides of the OFDM symbol in the frequency domain, at even sub-carrier spacings from 2 to 100 and 156 to 254, L denotes the index of the first pilots in second half. In the case of IEEE 802.16, L equals 156. Each value, $V_{\tilde{\epsilon}}$, can be computed simultaneously and separately using a multi-add accumulation scheme. The challenge in implementation is find a way to efficiently share resources when computing $V_{\tilde{\epsilon}}$. The pilots that are used to compute the correlation arrive every four cycles so there are four spare cycles between two consecutive computed pilots, allowing one multiply accumulate block to sequentially compute 4 separate correlations.

There are 8 sets of $A_{\tilde{\epsilon}}$ corresponding to 8 possible IFOs. These sets of $A_{\tilde{\epsilon}}$ can be pre-computed and stored separately. Thanks to the spreading of the computed pilots, the $A_{\tilde{\epsilon}}$ sets have many identical pilots – this naturally allows sharing between pre-rotated pilot sets – so that only 64 memory locations are required instead of 400. Thus, an 84% reduction in the

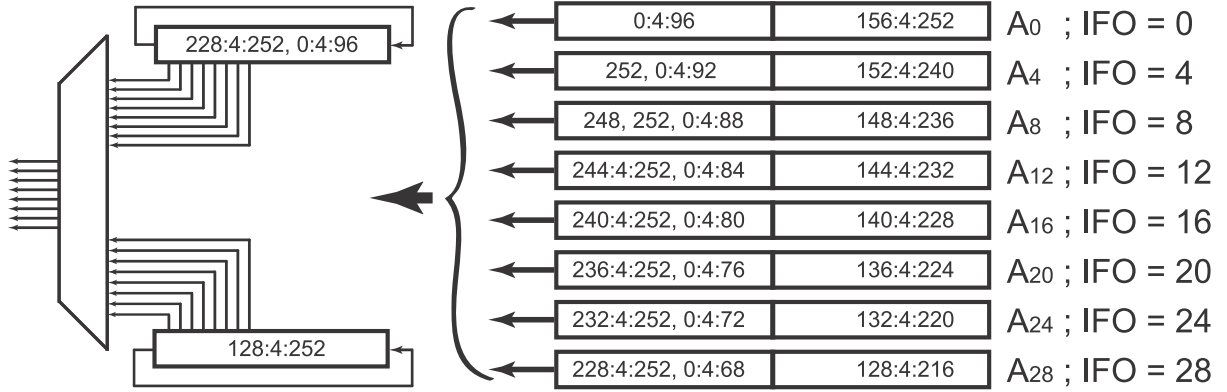


Figure 5.3: Circuit of known pilots shift register

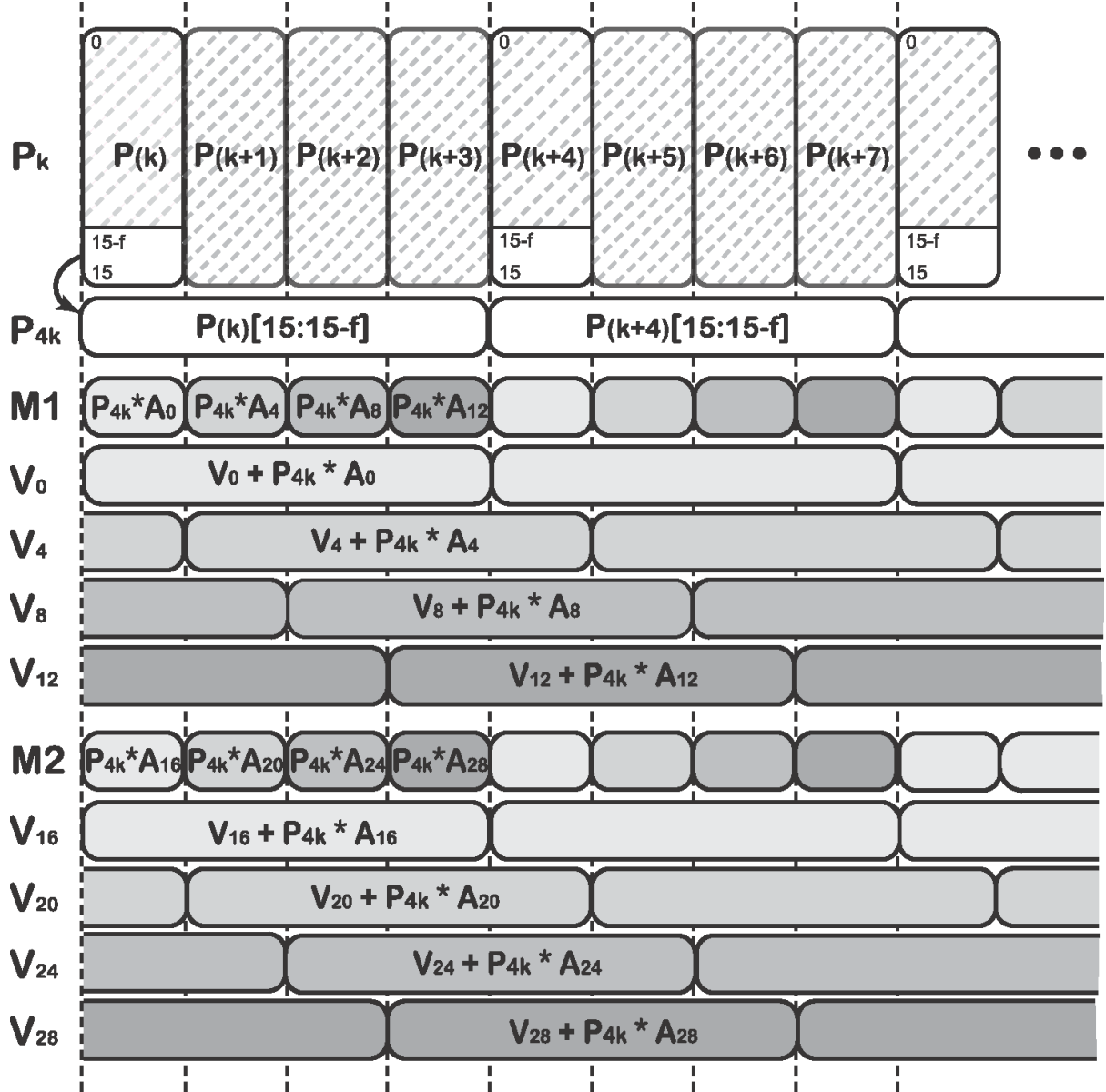
memory used to store $A_{\tilde{\epsilon}}$ sets is achieved. Fig. 5.3 illustrates the $A_{\tilde{\epsilon}}$ sets and circuitry for combining all $A_{\tilde{\epsilon}}$ sets.

Multiply accumulate blocks are shared for computing four sequential $V_{\tilde{\epsilon}}$ values over four successive clock periods. Fig. 5.4 demonstrates how this is done. P_k is received in every clock cycle. P_{4k} is the subsampling of P_k , taking a subset of the most significant bits from P_k every four cycles. The cross-correlation is performed with the values of P_{4k} . Two multipliers, $M1$ and $M2$ are used to compute the values of 8 cross-correlations $V_{\tilde{\epsilon}}$ in parallel. Each multiplier performs multiplications sequentially between P_{4k} and the corresponding transmitted pilots in 4 sets of $A_{\tilde{\epsilon}}$. The products are accumulated to the values of $V_{\tilde{\epsilon}}$. When all pilots are computed, the maximum operation, $\argmax|V|$, is performed on 8 $V_{\tilde{\epsilon}}$ values to estimate the IFO.

To design an optimised architecture for IFO estimation, the multiply-add formula of $V_{\tilde{\epsilon}}$ is mathematically manipulated into what is effectively a multiply-accumulate form. When one received sample is taken, $V_{\tilde{\epsilon}}$ can be expressed in the form of accumulation as

$$\begin{aligned}
 V_{\tilde{\epsilon}}(n) &= A_{\tilde{\epsilon}}(n)P(n) + V_{\tilde{\epsilon}}(n-1), \\
 &= (\Re\{c\} - i\Im\{c\})(\Re\{P(n)\} + i\Im\{P(n)\}) \\
 &\quad + (\Re\{V_{\tilde{\epsilon}}(n-1)\} + i\Im\{V_{\tilde{\epsilon}}(n-1)\})
 \end{aligned} \tag{5.6}$$

where $\Re\{\cdot\}$ and $\Im\{\cdot\}$ denote the real and imaginary parts, respectively, and $P(n)$, $A_{\tilde{\epsilon}}(n)$ denote the current values of the correlation of two consecutive received pilots and corresponding transmitted pilots, respectively. $V_{\tilde{\epsilon}}(n-1)$ is the accumulated value of $V_{\tilde{\epsilon}}$. To employ multiplierless correlation, $A_{\tilde{\epsilon}}$ are normalized to values c whose real and imaginary parts have values


 Figure 5.4: Resource sharing approach for computing $V_{\hat{\epsilon}}$.

in $\{-1, 0, 1\}$, and the wordlength of $P(n)$ and $V_{\tilde{\epsilon}}$ in fixed point format can be adjusted to increase estimation accuracy at the cost of increased hardware resource consumption. The real and imaginary parts of $V_{\tilde{\epsilon}}$ can be computed as

$$\begin{aligned}\Re\{V_{\tilde{\epsilon}}(n)\} &= \Re\{c\}\Re\{P(n)\} + \Im\{c\}\Im\{P(n)\} \\ &\quad + \Re\{V_{\tilde{\epsilon}}(n-1)\}, \\ \Im\{V_{\tilde{\epsilon}}(n)\} &= \Re\{c\}\Im\{P(n)\} - \Im\{c\}\Re\{P(n)\} \\ &\quad + \Im\{V_{\tilde{\epsilon}}(n-1)\}\end{aligned}\tag{5.7}$$

The proposed resource sharing architecture for the IFO estimator is shown in Fig. 5.5. The *ArgMax* module finds the maximum of the 8 $V_{\tilde{\epsilon}}$ values in order to identify the corresponding IFO estimate. The novelty of this IFO estimator is in terms of algorithmic improvements and architecture optimisation. This is achieved by firstly reformulating the equation into an expression that is computable with efficient shared resource circuitry. Thanks to the significant hardware reduction achieved, this IFO estimator can be feasibly implemented on a low-power, limited hardware resource FPGA, while simultaneously ensuring that performance is maintained by a trade off between number of pilots against word length.

We are able to demonstrate that the algorithm and structure optimisations mentioned above retain competitive estimation accuracy compared to conventional approaches, while also offering significant reductions in hardware resource usage. This makes it possible to implement a high-performance OFDM receiver on a low-power FPGA that has a limited number of DSP blocks available.

5.2.3 Simulation

Many variants of the proposed method were simulated in MATLAB using different channel models and the parameter set of the IEEE 802.16-2009 downlink preamble. Performance of the implementation was compared to the theoretical performance of some state of the art methods. This was assessed primarily in terms of the probability of failed estimation (POFE) with respect to channel SNR. POFE, which is widely used to evaluate the performance of IFO estimation [6, 10, 11], measures the number of fail estimations divided by the total number of IFO estimations. Overall, 100,000 IFO estimations were simulated in AWGN and Stanford

University Interim (SUI) [12] frequency selective channels. IFO estimation is performed with non-ideal FFO compensation, and FFO is determined and compensated using the method of Kim and Park [4]. The simulation also verifies the performance of the proposed method under the effect of RTO caused by imperfect STO estimation (assuming that STO estimation is still within the CP and does not cause ISI). In addition, a randomly generated amount of STO is added in the range from 0 to $N_{CP} - L - 1$, where L is the length of CIR.

We first investigate the performance degradation compared to theoretical performance as a result of reducing the number of pilots as proposed. Next, we will investigate the effect of wordlength optimisation. In both cases, comparisons are made with established methods in the literature that can be simulated but are otherwise infeasible for hardware implementation, namely the conventional method in [6] (*PCH*) that is applied to one training block with 100 pilots. In addition, two state of the art methods are also simulated for comparison. Firstly, metric *SY* from [10] as defined by,

$$\mu_{SY}(\tilde{\epsilon}) = \Re \left\{ \sum_{k=1}^{\frac{N}{2}} Y_{(2k-2)}^* Y_{(2k)} X_{(2k-\tilde{\epsilon})}^* X_{(2k-2-\tilde{\epsilon})} \right\} \quad (5.8)$$

where $\hat{\epsilon} = \underset{\tilde{\epsilon} \in S_{IFO}}{\operatorname{argmax}} \{ \mu(\tilde{\epsilon}) \}$. Secondly, metric *MM* from [11],

$$\mu_{MM}(\tilde{\epsilon}) = \Re \left\{ e^{i\frac{\pi}{4}} \sum_{k=1}^{\frac{N}{2}} Y_{(2k-2)}^* Y_{(2k)} X_{(2k-\tilde{\epsilon})}^* X_{(2k-2-\tilde{\epsilon})} \right\} \quad (5.9)$$

where $\Re\{\cdot\}$ denotes the real part.

We are unaware of any published circuits for these methods, because of their complex computation. The very large hardware requirement of the respective metrics does not lend these methods to feasible implementation on a low cost, low power FPGA (unlike the proposed method).

Performance Comparison

The performance of the proposed method, denoted *Prop*, is evaluated in comparison to the theoretical performance of state of the art methods by Park et. al. [6], Shim and You [10] and Morelli and Moretti [11], denoted *PCH*, *SY* and *MM*, respectively in the previous subsection. The theoretical performance is computed with full precision using full multiplication.

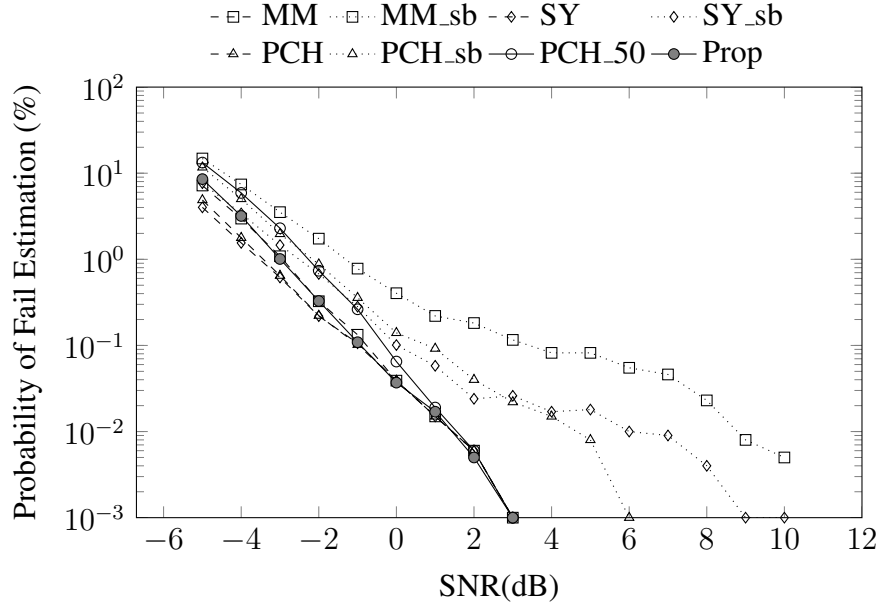


Figure 5.6: Fail rate of IFO estimation methods in AWGN channel without RTO.

However, it must be noted that implementing this directly in hardware would be prohibitive due to the large number of multiplication operations needed. In fact, hardware implementation would conventionally use sign-bit correlation instead of full precision correlation, as mentioned previously. Thus the full multiplication results shown here are undoubtedly better than those achievable in practice, and thus can be considered as upper performance bounds. For more realistic data, we also provide results from sign-bit correlation versions of the above, denoted *PCH_sb*, *SY_sb*, and *MM_sb* respectively.

The proposed method, evaluated against these, uses 50 spread pilots with indices that are multiples of 4. For the sake of comparison, an additional implementation of *PCH* is reported which, like the proposed method, uses 50 continuous pilots. This is denoted *PCH_50*. Figs. 5.6 and 5.7 plot performance results for all methods in an AWGN channel, with and without RTO respectively, and reveal that the proposed method generally performs well, especially at higher SNRs. Considering more realistic channel models, Figs. 5.8 and 5.9 plot the performance of all methods in SUI1, and SUI2 channels respectively, and similarly show that the proposed method performs well, especially at higher SNRs.

Under these experimental conditions, *PCH* and *SY* achieve equivalent performance in AWGN without RTO and in SUI1 channels. However, *SY* degrades more drastically than

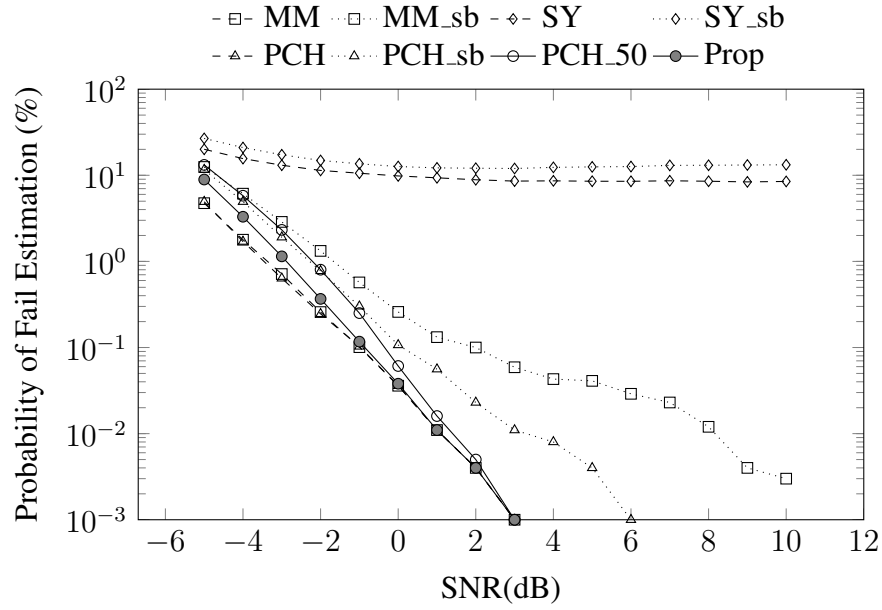


Figure 5.7: Fail rate of IFO estimation methods in AWGN channel with RTO.

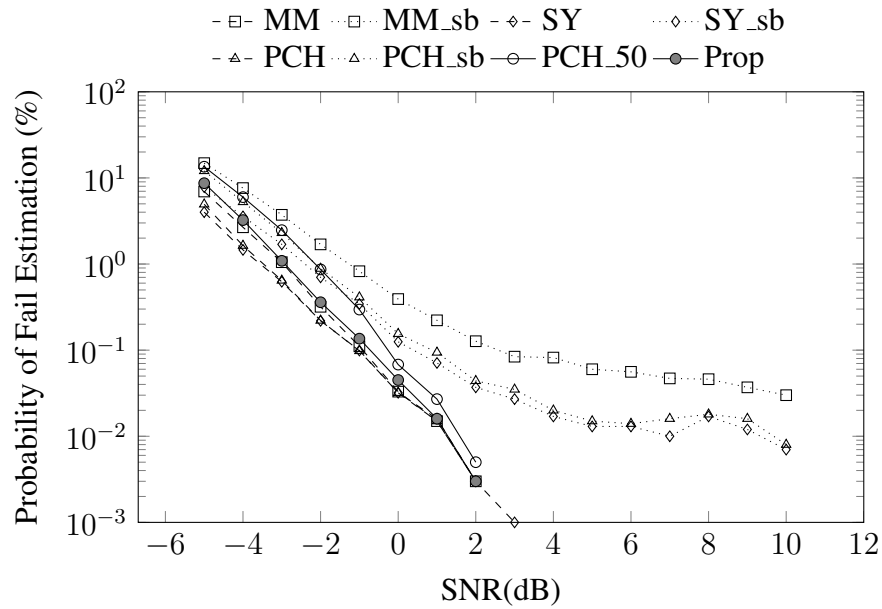


Figure 5.8: Fail rate of IFO estimation methods in SUI1 channel.

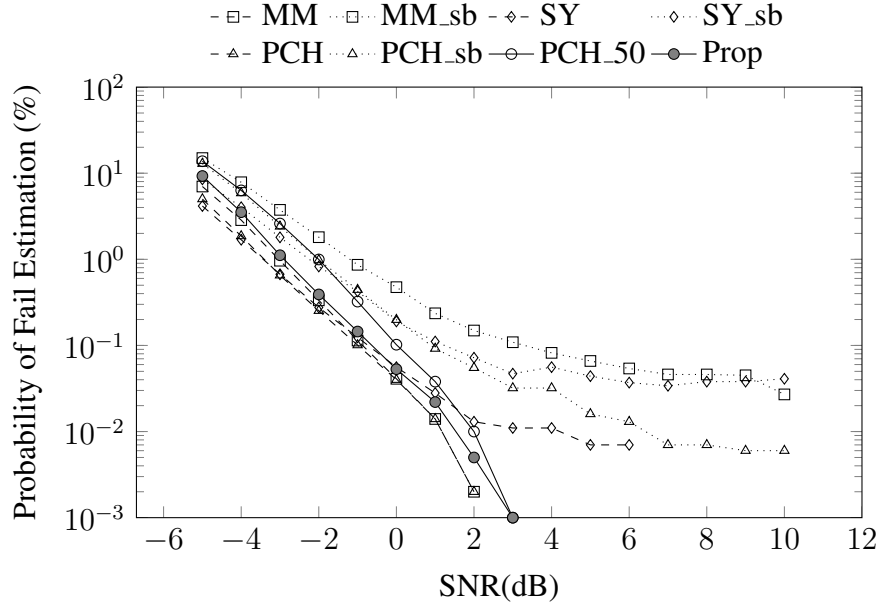


Figure 5.9: Fail rate of IFO estimation methods in SUI2 channel.

PCH in the SUI2 channel at SNRs above 1 dB. *SY* also deteriorates in the case of AWGN with RTO. This method appears to be very sensitive to large RTO, while *MM* and *PCH* exhibit better robustness to RTO. The accuracy of *MM* is slightly lower than that of *PCH* at SNRs below 0 dB, while performance is very similar at larger SNRs. Also note the performance of the conventional approach implementations, *PCH_sb*, *SY_sb*, *MM_sb* which degrade significantly with SNR, especially in the SUI1, SUI2 channels.

Apart from at very low SNRs, the proposed method, *Prop*, achieves almost identical performance to the simulated upper bound *PCH*, even in the presence of RTO. It should be noted that *Prop* achieves this while allowing the use of resource sharing through sparse pilot computation. This will be used to achieve a significant hardware saving. According to simulation results, *Prop*, with its spread pilots, is also more accurate than *PCH_50* that uses the same number of pilots spread continuously.

Wordlength Optimisation

Since sign-bit correlation degrades IFO estimation performance, specially in frequency selective channels, the proposed method instead employs multiplierless correlation to improve accuracy and robustness. The complexity of this approach is dependent on the wordlength

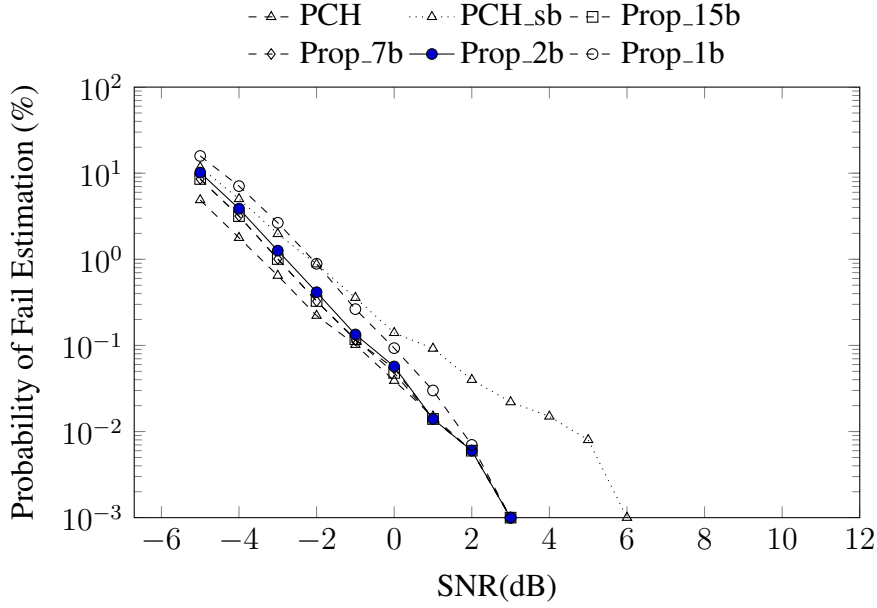


Figure 5.10: Fail rate for different wordlengths in AWGN channel without RTO.

chosen for the correlation computation. We now investigate how wordlength affects the performance of the proposed implementation, again compared to the theoretical bound, PCH , as well as to the performance of a conventional sign-bit implementation, PCH_sb . We denote wordlength using the notation $Q1.f$, meaning a single integer bit and f fractional bits. Evaluations are performed for f of 1, 2, 7, and 15 bits. These results will be plotted in the following figures with the labels $Prop_1b$, $Prop_2b$, $Prop_7b$, and $Prop_15b$, respectively. Figs. 5.10 and 5.11 plot the performance in AWGN (with and without RTO), with all tested wordlengths in the proposed method performing comparably to PCH (and being much better than PCH_sb at SNRs exceeding about 2 dB). Figs. 5.12 and 5.13 plot the results when using the more realistic SUI1 and SUI2 channel models.

It can be seen from the plots that each of the tested wordlengths achieves much better performance and exhibits greater robustness to frequency selective channels than the sign-bit realisation of the conventional approach, PCH_sb . Additionally, these realisations of the proposed method do not suffer as much degradation in the presence of RTO. Moreover, it is possible to improve low SNR performance by adopting a longer wordlength with the proposed method, at a cost of increased hardware complexity. Increasing wordlength does improve results slightly, with the step from 1 to 2 bits being the most significant gain. By contrast, increasing from

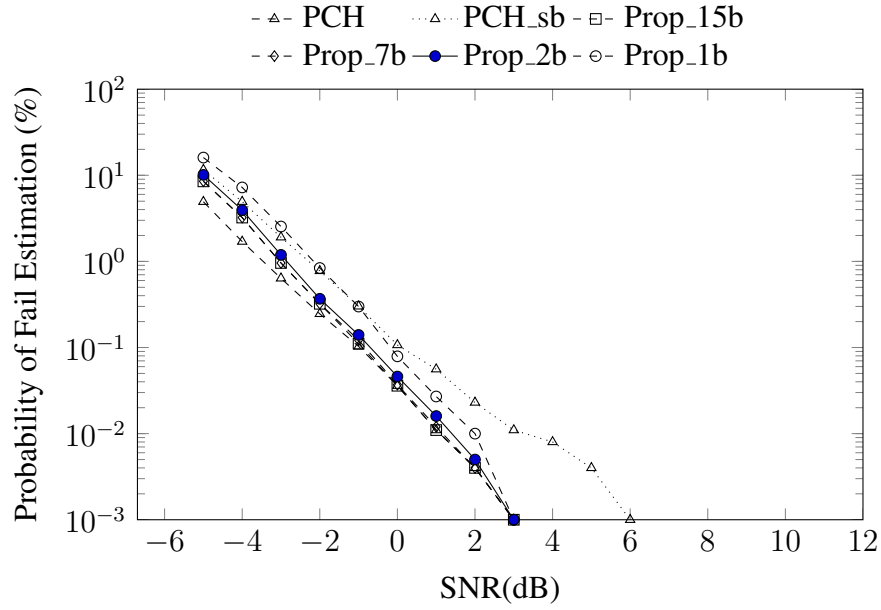


Figure 5.11: Fail rate for different wordlengths in AWGN channel with RTO.

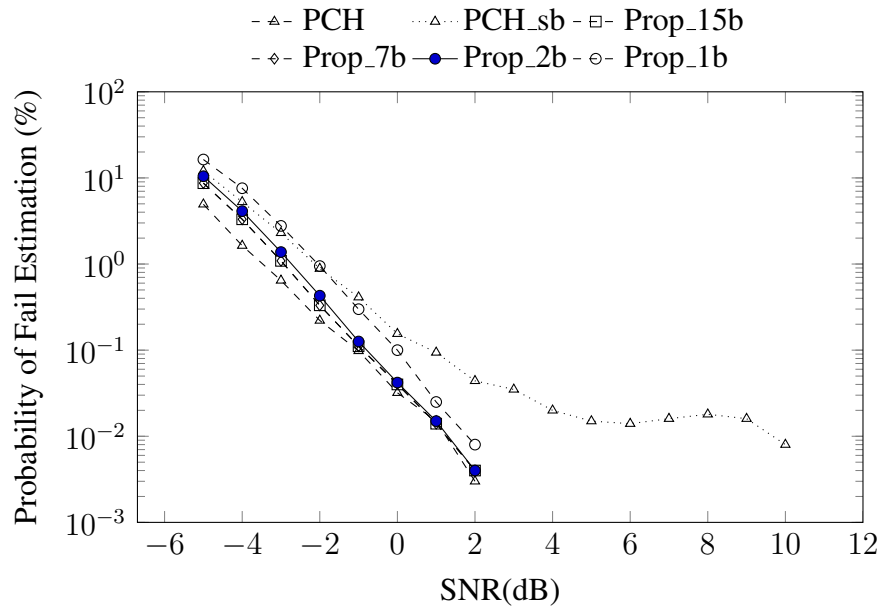


Figure 5.12: Fail rate for different wordlengths in SUI1 channel.

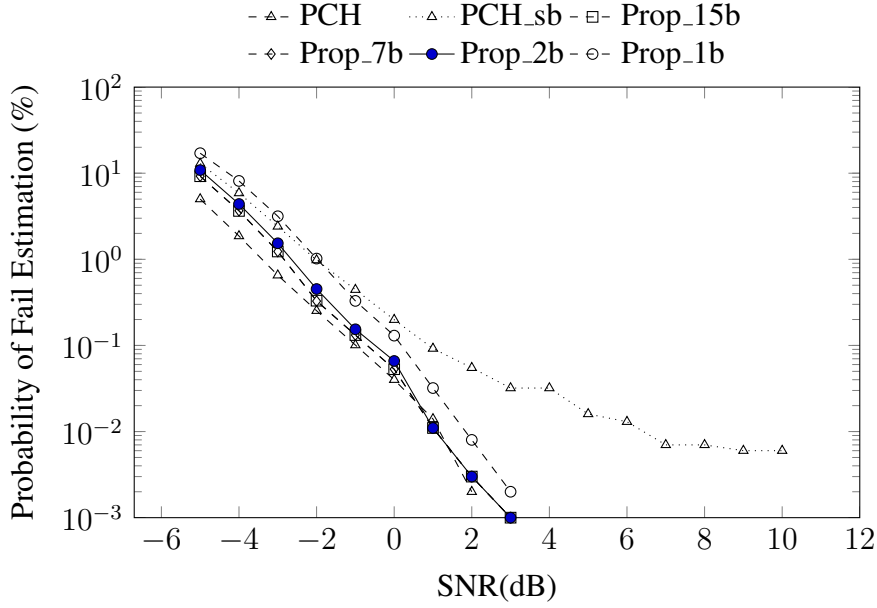


Figure 5.13: Fail rate for different wordlengths in SUI2 channel.

2 to 7 or from 7 to 15 bits has little impact. In general, *Prop_2b* achieves an estimation accuracy close to that of the theoretical performance bound, *PCH*, at intermediate and higher SNRs, even though it involves computation with fewer bits, and can hence be implemented more efficiently.

5.2.4 FPGA Implementation

The analysis in Section 5.2.3 suggests that the proposed method offers comparable estimation performance to existing methods in the literature. As a result of the simplifications inherent in the proposed approach, this should be achievable at a reduced hardware cost. This section now quantifies this hardware cost, for an FPGA-based implementation. It is important to note that these hardware savings are accessible for a number of target implementation devices, although we are interested primarily in FPGA implementation as part of our work on leveraging FPGA reconfigurability in cognitive radios.

Conventional Approach

To obtain the theoretical performance previously discussed in Section 5.2.3 and denoted as *PCH*, the computation of the estimation metric in [6], using 100 pilots, would require about

200 complex multipliers, resulting in the use of over 600 DSP blocks. This may exceed the available resources on small devices, or leave insufficient resources for other tasks on larger devices. As the number of multiplications required for a full implementation of the approach is prohibitive, the conventional approach for implementation, as we have discussed, uses sign-bit correlation [4]. This conventional implementation uses all 100 pilots in the long preamble to perform sign-bit correlation, and multiply_adds are eliminated at taps where the pilots of the long preamble are not used. This implementation mirrors the *PCH_sb* in Section 5.2.3, and allows us to quantify the benefits of our proposed approach against a known reference benchmark.

Proposed Approach

The proposed architecture for IFO estimator implemented with several different wordlengths of $P(n)$ and V_{ϵ} in (5.6) are compared, to allow us to explore the hardware costs associated with the respective implementations. Four fixed point formats for $P(n)$, are investigated: Q1.1, Q1.2, Q1.7, and Q1.15. V_{ϵ} is represented accordingly in Q7.1, Q7.2, Q7.7, and Q7.15 formats to avoid overflow.

In order to obtain a comprehensive optimised implementation, these circuits are each implemented using two different structures. The first uses only logic elements (LE) for computation, while the second uses Xilinx DSP48A1 [13] primitives. Considering (5.7), $\Re\{V_{\epsilon}(n)\}$, $\Im\{V_{\epsilon}(n)\}$ can be computed effectively using two DSP blocks as 3-input adders, instead of 4 blocks as would be usual. Fig. 5.14 illustrates how this is done for $\Re\{V_{\epsilon}(n)\}$, and similarly for $\Im\{V_{\epsilon}(n)\}$. Note that the solution presented in Fig. 5.14 is optimised for QPSK modulated pilots (since their amplitudes are identical) as specified in IEEE 802.16, as well as in most OFDM-based standards. The normalisation performed in (5.7) allows the correlation to be reduced to two DSP blocks operating as 3-input adders (instead of 4 DSP blocks with multipliers as would be usual). These methods correspond to *Prop_1b*, *Prop_2b*, *Prop_7b*, and *Prop_15b* that were investigated for estimation accuracy in Section 5.2.3.

5.2.5 Implementation Results

The circuits were synthesised and fully implemented using Xilinx ISE 13.2, targeting the low-power Xilinx Spartan-6 XC6SLX75T FPGA. The results are reported in terms of the number

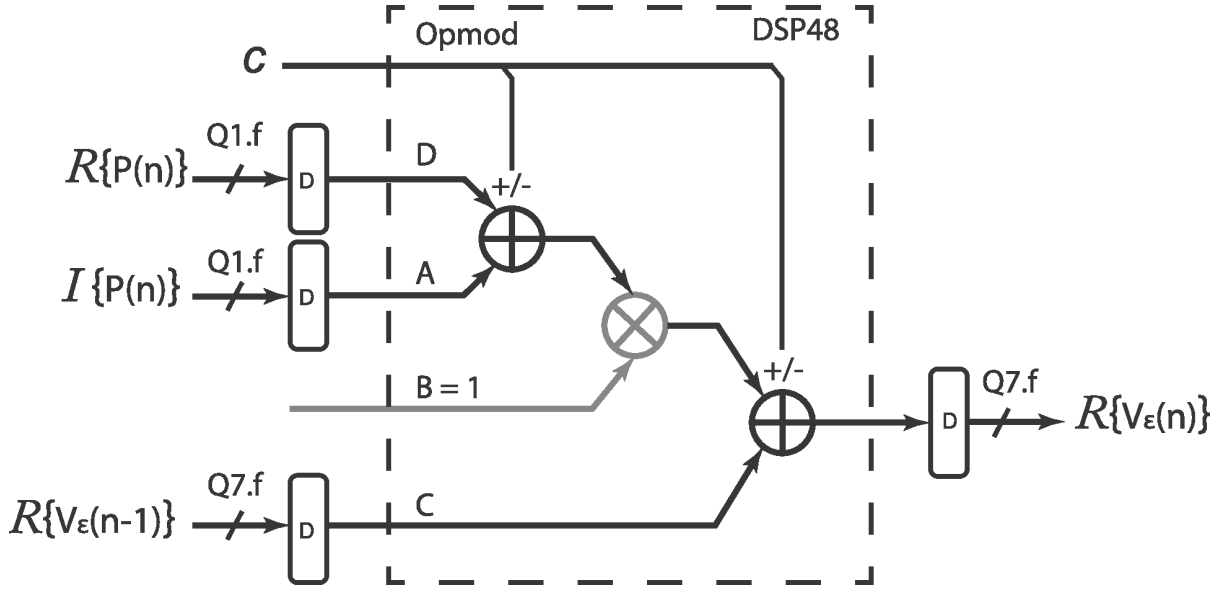


Figure 5.14: DSP block based 3-input adder for correlation.

of flip-flops (FF), look-up tables (LUT), and DSP48 blocks, along with dynamic power consumption, as summarised in Table 5.1.

Table 5.1: Resource utilisation and dynamic power consumption of IFO estimators.

| IFO est. Cir. | #FF | #LUT | #DSP | Pre (MHz) | D. Power |
|-------------------|-----------|-----------|------|-----------|----------|
| conv_100p | 3270 (3%) | 1837 (3%) | 3 | 142 | 42 mW |
| Prop_1b_LE | 328 (1%) | 370 (1%) | 3 | 136 | 9 mW |
| Prop_2b_LE | 350 (1%) | 390 (1%) | 3 | 136 | 10 mW |
| Prop_7b_LE | 460 (1%) | 471 (1%) | 3 | 136 | 12 mW |
| Prop_15b_LE | 735 (1%) | 696 (1%) | 3 | 134 | 17 mW |
| Prop_1b_DSP | 328 (1%) | 306 (1%) | 7 | 78 | 11 mW |
| Prop_2b_DSP | 350 (1%) | 319 (1%) | 7 | 78 | 12 mW |
| Prop_7b_DSP | 460 (1%) | 379 (1%) | 7 | 77 | 14 mW |
| Prop_15b_DSP | 735 (1%) | 591 (1%) | 7 | 77 | 18 mW |

conv_100p refers to the conventional approach, implemented using sign-bit correlation over 100 pilots. *Prop_fb_LE*, *Prop_fb_DSP*, in which $f = 1, 2, 7$ and 15 (corresponding to received sample format Q1.f), denote the circuits of corresponding wordlengths, implemented using

logic elements and DSP48 blocks, respectively. Referring to the table, the proposed implementation shows that significant improvement in resource usage and dynamic power consumption is possible with the proposed method.

The hardware resources used by *Prop_fb_LE* and *Prop_fb_DSP* increase gradually, in terms of FFs and LUTs, as the wordlength increases. The number of FFs used in *Prop_fb_DSP* and *Prop_fb_LE* is equal, while *Prop_fb_DSP* uses fewer LUTs since the DSP blocks are used for the 3-input additions.

The *Prop_fb_LE* implementations use 3 DSP blocks to compute $P(n)$, while *Prop_fb_DSP* require an additional 4 DSP blocks to perform the correlation. *Prop_fb_LE*, *Prop_fb_DSP* both consume far fewer LUT and FF resources than the conventional *conv_100p* implementation.

For *Prop_2b_LE*, the number of FFs and LUTs is reduced by 90% and 79% respectively compared to the conventional *conv_100p* approach.

The maximum frequencies of circuits, reported after place and route, comfortably exceed the timing requirements for IEEE 802.16 synchronisation whose sampling frequency is below 25 MHz. A post-place-and-route simulation was used to estimate the power consumption of the system at a clock rate of 50 MHz using the Xilinx XPower tool – also shown in table 5.1.

Prop_fb_LE implementations consume less power than the equivalent *Prop_fb_DSP* implementations. All implementations of the proposed method consume significantly less power than the conventional implementation. *Prop_2b_LE* consumes just 22% of the power consumed by *conv_100p*.

In subsection 5.2.3, we established that *Prop_2b_LE* easily outperforms the conventional approach in terms of estimation accuracy. In this subsection, we have shown that it does so with a significant hardware resource saving, and with significantly reduced power consumption. In fact, the estimation performance of *Prop_2b_LE*, in AWGN and SUI channels (except at very low SNR), is close to the theoretical bound of *PCH*, which would demand a significant proportion of the FPGAs resources if it were implemented conventionally. Meanwhile, *Prop_2b_LE* is extremely efficient, consuming less than 1% of the resources available on a low-power Spartan-6 XC6SLX75 FPGA. Beyond IEEE 802.16-2009, the folded resource sharing architecture, which leverages sub-sample spaced OFDM pilots, can be adopted for use in other OFDM standards (including IEEE 802.11 and IEEE 802.22).

5.3 Summary

This chapter investigated IFO estimation in OFDM-based systems such as IEEE 802.16. A technique is proposed for efficient implementation of IFO estimation, which aims in particular for a low power and low resource utilisation. Since IFO estimation contributes significantly to the complexity of a robust synchroniser design, this work is important for multistandard radios, or applications where significant frequency variation is expected. Robust IFO estimation allows for relaxed analogue RF constraints, leading to reduced cost. A modified timing metric is derived which allows for resource sharing to reduce both resource requirements as well as power consumption. The proposed implementation makes use of a four-fold resource sharing architecture to significantly reduce hardware cost, while multiplierless correlation with optimised wordlength is used to improve estimation accuracy in comparison to a conventional implementation approach using signbit correlation. The method is shown to perform as well as current state-of-the-art methods that employ multiplier-based correlation, and yet with significantly improved power and resource requirements. Dynamic power consumption is reduced by 78% over even a sign-bit version of the conventional approach, yet it offers better estimation performance in both AWGN and frequency selective channels. Beyond IEEE 802.16-2009, the folded resource sharing method, which leverages sub-sample spaced OFDM pilots, can be adopted for use in other OFDM standards (including IEEE 802.11 and IEEE 802.22). Multiplierless correlation with word length optimisation is already used within communication systems, however the trade off between the number of pilots and computational accuracy for timing synchronisation is relatively unexplored in the research literature. This work has been submitted as a journal paper [J3].

Chapter 6

A Spectrum Efficient Shaping Method

Chapter 7

A Novel Architecture for Multiple Standard Cognitive Radios

Chapter 8

Conclusion and Future Work

References

- [1] T. Schmidl and D. Cox, “Robust frequency and timing synchronization for OFDM,” *IEEE Transactions on Communications*, vol. 45, pp. 1613 –1621, Dec. 1997.
- [2] C. N. Kishore and V. U. Reddy, “A frame synchronization and frequency offset estimation algorithm for OFDM system and its analysis,” *EURASIP Journal on Wireless Communications and Networking*, vol. 2006, pp. 1–16, 2006.
- [3] K. Bang, N. Cho, J. Cho, H. Jun, K. Kim, H. Park, and D. Hong, “A coarse frequency offset estimation in an OFDM system using the concept of the coherence phase bandwidth,” *IEEE Transactions on Communications*, vol. 49, pp. 1320 –1324, Aug. 2001.
- [4] T.-H. Kim and I.-C. Park, “Low-power and high-accurate synchronization for IEEE 802.16d systems,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 16, pp. 1620 –1630, Dec. 2008.
- [5] T. H. Pham, I. V. McLoughlin, and S. A. Fahmy, “Robust and Efficient OFDM Synchronisation for FPGA-Based Radios,” *Circuits, Systems, and Signal Processing*, vol. 33, pp. 2475–2493, Aug. 2014, Springer.
- [6] M. Park, N. Cho, J. Cho, and D. Hong, “Robust integer frequency offset estimator with ambiguity of symbol timing offset for OFDM systems,” in *Proc. Vehic. Techn. Conf. (VTC)*, pp. 2116–2120, 2002.
- [7] L. Schwoerer, “VLSI suitable synchronization algorithms and architecture for IEEE 802.11a physical layer,” in *IEEE International Symposium on Circuits and Systems, 2002. ISCAS 2002*, vol. 5, pp. V–721, 2002.

REFERENCES

- [8] IEEE std.802.16-2009, *IEEE Standard for Local and Metropolitan Area Networks Part16: Air Interface for Fixed Broadband Wireless Access Systems*.
- [9] T. H. Pham, S. A. Fahmy, and I. V. McLoughlin, “Low-power correlation for IEEE 802.16 OFDM synchronisation on FPGA,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 21, pp. 1549–1553, August 2013.
- [10] E.-S. Shim and Y.-H. You, “OFDM integer frequency offset estimator in rapidly time-varying channels,” in *Asia-Pacific Conf. on Commun.*, pp. 1–4, 2006.
- [11] M. Morelli and M. Moretti, “Integer frequency offset recovery in OFDM transmissions over selective channels,” *IEEE Transactions on Wireless Communications*, vol. 7, pp. 5220–5226, Dec. 2008.
- [12] V. Erceg, K. V. S. Hari, and M. S. Smith, “Channel models for fixed wireless applications,” *Tech. Rep. IEEE 802.16a-03/01*, Jul. 2003.
- [13] Xilinx Inc., *UG389: Spartan-6 FPGA DSP48A1 Slice*, August 2009.