

An Ultralow Power System on Chip for Automatic Sleep Staging

Syed Anas Imtiaz, *Member, IEEE*, Zhou Jiang, and
Esther Rodriguez-Villegas, *Senior Member, IEEE*

Abstract—This paper presents an ultralow power system on chip (SoC) for automatic sleep staging using a single electroencephalogram (EEG) channel. The system integrates an analog front end for EEG data acquisition and a digital processor to extract spectral features from these data and classify them into one of the sleep stages. The digital processor consists of multiple blocks implementing an automatic sleep staging algorithm that uses a set of contextual decision trees controlled by a state machine. The processor is designed to stay in the idle mode at most times waking up only when computations are required. In addition, the mathematical operations are implemented in a way such that the number of datapath components needed is very small. The SoC is implemented in an AMS 0.18- μm CMOS technology and is powered using a single 1.25-V supply. Its power consumption is measured to be 575 μW , while its classification accuracy using real EEG data is 98.7%.

Index Terms—Electroencephalogram (EEG), electroencephalography, low-power biomedical system, sleep classification algorithm, sleep staging.

I. INTRODUCTION

SLEEP is a natural state of reduced alertness during which the response of the human body to external stimuli decreases. It is considered a necessity of life for humans and animals alike and is essential to their physical and emotional wellbeing. Sleep accounts for approximately one-third of our lifetime. However, it is not only the number of hours that defines a healthy sleep but also its composition and architecture.

Human sleep is broadly classified into two distinct oscillatory phases based on the eye movements during sleep: rapid eye movement (REM) and nonrapid eye movement (NREM). According to the American Academy of Sleep Medicine (AASM) rules for sleep classification [1], the NREM phase is further divided into N1, N2, and N3 stages based on the depth of sleep. Hence, together with REM (R) and Wake (W), there are a total five well-defined sleep stages that can be identified from the electroencephalogram (EEG), electrooculogram (EOG), and electromyogram (EMG) signals

during sleep. Analysis of these stages, their timing, duration, and occurrence rate are extremely helpful in investigating sleep-related complaints.

In this paper, a fully integrated ultralow power SoC is presented to acquire data from just a single EEG channel and perform automatic sleep staging. This integrated circuit (IC) can be used to create a wearable EEG-based sleep monitoring system. The following section presents an overview of current sleep monitoring technologies with their limitations and challenges and explains how wearable devices can be used to overcome some of these challenges. Section III discusses the possible system architectures for the design of such wearable devices. Section IV briefly describes the sleep staging algorithm that has been developed for an IC implementation. In Section V, the design and circuit-level implementation of each constituent block of the algorithm is covered in detail. Finally, the SoC performance results and classification accuracy are discussed in Section VI.

II. SLEEP MONITORING AND POLYSOMNOGRAPHY

It is estimated that one-third of the global population suffers from abnormal sleep everyday with more than 10% having a clinically significant sleep problem [2]. The symptoms of sleep disorders may include sleep deprivation, disruptive sleep, excessive sleepiness, and other sleep abnormalities that can be fatal if left untreated [3], [4]. These result in social consequences for individuals with sleep problems and a collective financial impact on the economy due to expensive treatments, reduced productivity, accidents, as well as comorbidities resulting from delayed diagnosis [5].

Sleep disorders are generally presented to the doctors in the form of complaints about difficulties in falling asleep or experiencing unusual sleepiness during daytime. Initial assessments include patient history, interviews, and questionnaires that may be followed up with further tests and monitoring. Patients are normally asked to maintain diaries to log their sleeping habits for about two weeks. These details, together with other lifestyle information, help further the diagnosis and treatment. Although a sleep diary provides invaluable information about the sleeping habits of patients, it is very subjective and based on their perception of sleep. Further, in many cases, more observations and information about the sleep architecture are needed, which requires the use of different clinical tools.

A popularly used inexpensive and noninvasive method for sleep monitoring is actigraphy that involves a wrist-worn device to sense movements and record sleep/wake patterns over a long period of time. However, it lacks the ability

Manuscript received October 12, 2016; revised December 8, 2016; accepted December 30, 2016. Date of publication January 25, 2017; date of current version March 3, 2017. This paper was approved by Associate Editor Vivek De. This work was supported by the European Research Council through the European Community's 7th Framework Programme (FP7/2007-2013) under Grant 239749.

The authors are with the Circuits and Systems Group, Electrical and Electronic Engineering Department, Imperial College London, London SW7 2AZ, U.K. (e-mail: anas.imtiaz@imperial.ac.uk; zhou.jiang11@imperial.ac.uk; e.rodriguez@imperial.ac.uk).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/JSSC.2017.2647923

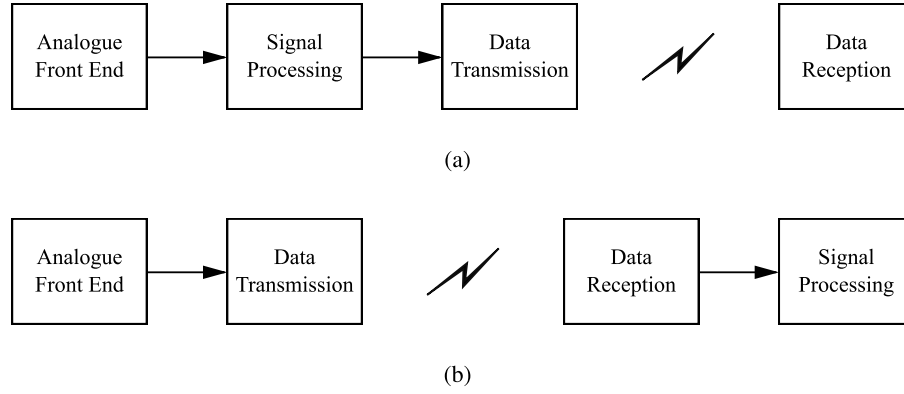


Fig. 1. Two system design approaches for a wireless wearable device. (a) With signal processing at the sensor node. (b) With signal processing at the receiver end.

to provide sleep staging information that is diagnostically vital in many cases [6]. To obtain this, patients are asked to attend a sleep clinic where their sleep is monitored in a test known as polysomnography (PSG). This is the gold standard of sleep monitoring during which brain activity (EEG), eye movements (EOG), and muscle movements (EMG) are monitored together with oxygen saturation, air flow, respiratory effort, and others (if needed). The EEG, EOG, and EMG signals are manually scored by sleep technicians in blocks of 30-s epochs based on the AASM rules of classification. The resulting diagnostic graph depicting sleep stages as a function of time is known as a *hypnogram*.

PSG is not very comfortable for patients due to the large number of electrodes attached to them during their sleep. It often requires multiple nights of monitoring to allow time for them to acclimatize and to ensure sufficient data have been obtained. The subsequent manual analysis is also a tedious and error-prone task that can take up to four hours for scoring [7] with high disagreement rate among scorers [8], [9]. All of these make sleep monitoring a significantly time-consuming and costly process. The financial aspect is now more relevant than ever with rising population and soaring healthcare costs globally. As a result, active research is ongoing in the field of sleep medicine to solve some of these problems. To reduce data analysis time and inter-rater disagreements, intelligent algorithms are being developed for automatic sleep stage scoring. There is also an effort to reduce the number of recording channels and to use wireless sensors to make PSG comfortable for patients.

Traditionally, three EEG channels are required in PSG systems together with EOG and EMG channels. However, Ruehland *et al.* [10] reported no significant differences in sleep scoring reliability when using a single EEG channel. The EOG and EMG channels are also traditionally required as REM stage epochs are identified by observing the chin muscle and eye activity [1]. Our recent work [11], however, demonstrated reliable detection of REM sleep from a single EEG channel that allows using only this channel to perform full sleep staging.

While reducing the number of channels, using wireless sensors and adding automatic analysis are important steps

forward; in order for a sleep staging system to be practically useful, it must also be small in size and wearable so that the patients can use it with ease and comfort for long hours. At the same time, the analysis must provide reliable results for it to be clinically useful. A low-power sleep staging system on chip (SoC) can help to meet all these criteria leading to a truly wearable system. With wireless transmission from a single EEG channel and low-power automatic sleep staging, the overall system would be comfortable to use for patients, save analysis time, improve scoring reliability, and reduce overall costs. It could also record sleep parameters more accurately without the need of sleep logs that need to be filled in by patients.

III. WEARABLE SLEEP STAGING SYSTEM

A truly wearable device for automatic sleep staging would entail a wireless noninvasive sensor that transmits data to a small handheld device. Fig. 1 shows two possible architectures for the design of such a system where the only difference between the two is the placement of the signal processing block in the data pipeline.

The first approach uses signal processing at the sensor end so that only the sleep stage information is transmitted every 30 s, resulting in a very small data rate. This scenario is an ideal use case for ultralow power transmitters, such as bluetooth low energy [12] that are widely available in most smartphones. However, due to strict power constraints of the sensing device, its processing capability is very limited. Hence, a tradeoff between acceptable levels of performance and algorithm complexity must be made to meet system specifications. In the second approach, signal processing is performed at the receiver end. Thus, all raw data need to be transmitted. With much more relaxed constraints at the receiver, complex algorithms can be used to process the received data. However, using this approach, the transmission stage consumes more power due to a significantly higher data rate.

For both of these approaches, the receiver can either be a handheld device (smartphone, tablet, etc.) or a desktop computer, depending on the processing needs. While the first approach definitely requires low complexity sleep staging algorithms for the aforementioned reasons, limiting the

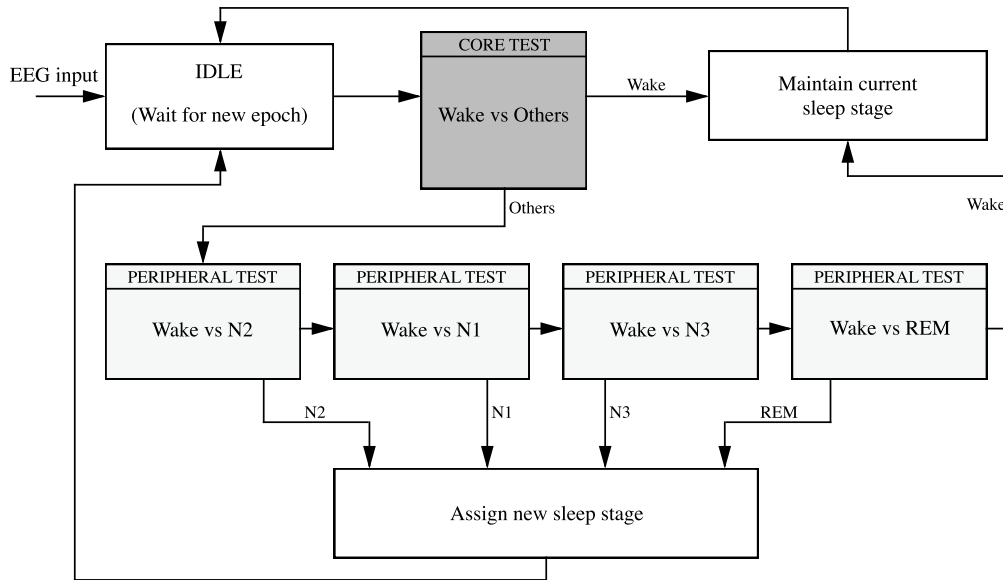


Fig. 2. Overview of the automatic sleep staging algorithm to be implemented on chip.

second approach to cases where the receiver is a smartphone also puts some constraints on algorithm development. This is because compared with a desktop computer or a server, a smartphone will have lower processing resources available. Further, keeping in mind the growing trend of mobile device usage, it is safe to say that at least from the perspective of user experience, it makes sense to have a system that uses these devices as a receiver. Hence, regardless of the chosen architecture, there is a definite need for low-complexity sleep staging algorithms that can be run on a sensor node or a smartphone. It is for this reason the sleep staging algorithm in [11] was developed. The rest of this paper focuses on the implementation of this algorithm as a complete SoC.

IV. AUTOMATIC SLEEP STAGING ALGORITHM

A number of sleep staging algorithms have been published in the literature over the past five decades, however very few have been implemented in hardware [13], [14]. Most of the reported algorithms have a good classification accuracy, which comes at the cost of computational complexity using a large number of features and classifiers. On hardware, this translates into higher power consumption. Since wearable systems have very limited power budget and computational resources, it is not feasible to run complex algorithms on such systems. Hence, in [11], a sleep staging algorithm was specifically developed to be used in wearable systems. This is a contextually aware algorithm that uses a set of spectral features and a combination of small decision trees to determine the next sleep stage based on the current stage.

Fig. 2 shows an overview of the sleep staging algorithm. It is based on the observation that when a certain sleep stage is prevalent, it *normally* stays for a few epochs before transitioning into the next stage. This means that the classifier only needs to determine whether an epoch is of the same sleep

stage or not using a *one-versus-all* decision tree. If a stage change is detected, a series of *one-versus-one* decision trees is used to determine the new stage. The decision trees, their order of execution, and the features needed are established based on the current sleep stage as well as the likelihood of the next one.

There are two levels of tests in this sleep staging algorithm: core tests and peripheral tests. A core test determines whether an epoch being analyzed is of the same sleep stage as the previous one or not. Since there are a total of five sleep stages, the algorithm consists of five core tests with only one of them active in a given state. If the active core test determines that the current epoch may be of a different sleep stage, then a series of peripheral tests is applied to find the new stage. These are very small one-versus-one decision trees that are executed in a specific order. If one of these tests is passed, the rest are not evaluated and the new sleep stage is assigned. In the event that all peripheral tests fail, the state of the machine remains unchanged and the current epoch is assigned the previous sleep stage. There can be a maximum of four peripheral tests corresponding to each core test since the classifier is looking for one of the other four stages except the current stage.

For example, in Fig. 2, the algorithm starts with a previous epoch scored as Wake (this is also the default power-on state). It first checks whether the current epoch is also to be classified as Wake or as one of the other sleep stages using the *Wake vs Others* core test. If its result is Wake, no change is needed and the state machine returns to an idle state. If, however, the result is others, the peripheral tests are used to determine the new state of the machine. For the Wake state, these peripheral tests are *Wake vs N1*, *Wake vs N2*, *Wake vs N3*, and *Wake vs REM*. These will determine the classification label for the current epoch and subsequently result in the state machine being transitioned into the corresponding state.

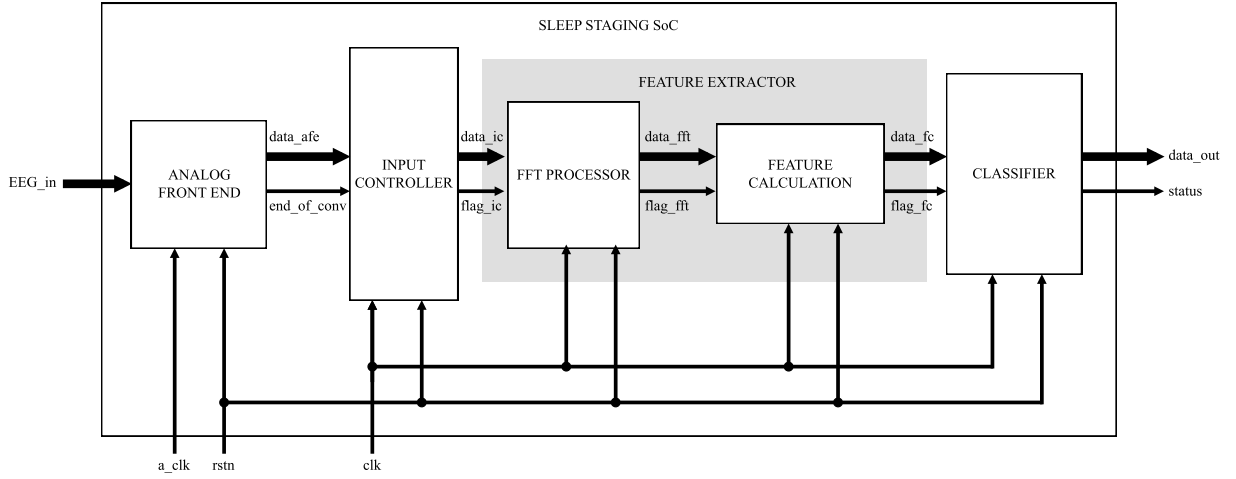


Fig. 3. Top-level block diagram of the automatic sleep staging SoC showing the main modules.

As a result, the next epoch will be classified by starting at a different core test following a similar pattern of peripheral tests.

V. SLEEP STAGING SoC ARCHITECTURE

Fig. 3 shows the four main blocks of the sleep staging SoC: 1) analog frontend; 2) input controller; 3) feature extractor; and 4) classifier. The function, architecture, and implementation details of these blocks are explained in the following sections.

A. Analog Frontend

The *analog frontend* consists of a low-power and low-noise neural amplifier connected to a 12-b successive approximation register analog-to-digital converter (ADC) [15]. It uses a clock frequency of 8192 Hz in order to achieve a sampling rate of 256 Hz. The amplifier utilizes an on-chip ultrahigh pseudoresistor ($>10^{10} \Omega$) to create a high-pass filter with an ultralow cutoff frequency (<0.5 Hz), thus rejecting large dc offsets at its input in a similar configuration to the one in [16]. The neural amplifier achieves a gain of 36 dB while consuming 13.8 μA of current.

B. Input Controller

The sleep staging algorithm processes data in blocks of 2-s subepochs and therefore requires 512 data samples to be buffered prior to any processing. This is performed in the *input controller*, which is simply a register bank with an address counter. As shown in Fig. 4, it reads the ADC output at each end of conversion and stores the value at the address pointed by the counter. When 512 samples are received, the counter reaches its maximum limit and is reset to zero. At the same time, a pulse is generated as an instruction for the next block to read the stored data and start computation. This frees up the register bank to continue receiving samples for the next subepoch. The *input controller* also includes circuitry to synchronize the end of conversion signal with the digital clock.

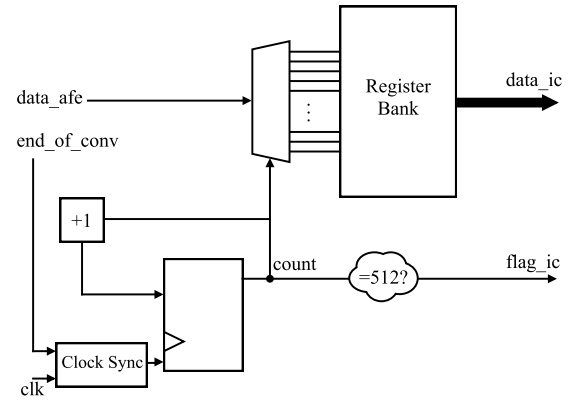


Fig. 4. Block diagram of the input controller.

C. Feature Extractor

The *feature extractor* is the computational heart of the sleep staging SoC where most of the mathematical operations take place. At its core is a fast Fourier transform (FFT) processor, which transforms the signal into frequency domain from which the spectral features are calculated. These features include the spectral edge frequencies and signal power ratios in different frequency bands. More details about these features can be found in [11].

1) *FFT Processor*: The FFT is an algorithm to speed up the computation of the discrete Fourier transform (DFT) by making use of its symmetrical properties. This greatly reduces the number of multiplications required, resulting in an algorithmic complexity of $O(N \log N)$ compared with $O(N^2)$ when using the original DFT. The FFT of an input signal x is computed as follows:

$$X(k) = \frac{1}{N} \sum_{n=0}^{N-1} x(n) e^{-j \frac{2\pi nk}{N}} = \frac{1}{N} \sum_{n=0}^{N-1} x(n) \omega_N^{nk} \quad (1)$$

where $X(k)$ is the transformed output at index k and N is the length of the signal. The most popular algorithm to compute

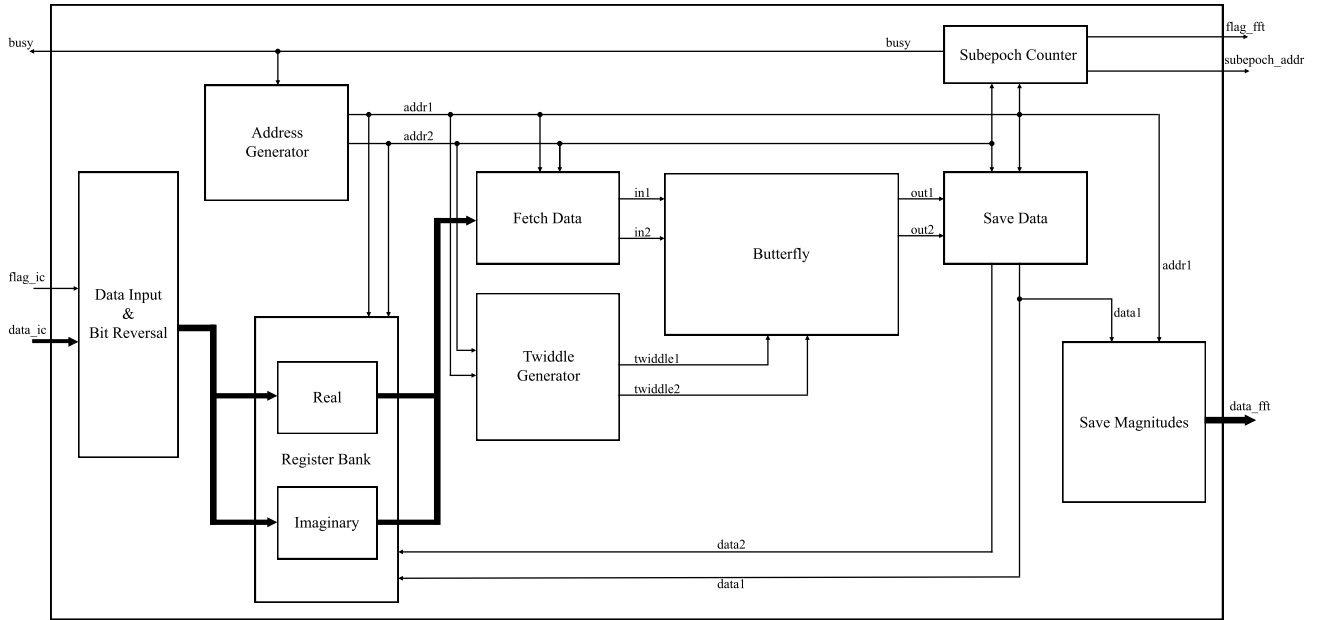


Fig. 5. Block diagram of the FFT processor.

the FFT is the Cooley–Tukey algorithm [17]. Its simplest variant is radix-2 decimation in time that breaks down the entire calculation into a number of 2-point DFTs. An N -point FFT requires $N/2$ 2-point DFT computations at each level, with a total of $n(= \log_2 N)$ levels. The mathematical operation involved in computing this DFT is often referred to as the *butterfly* operation and involves one complex multiplication and two complex additions. Hence, a total of $Nn/2$ such butterfly operations are needed to compute an N -point FFT. However, since all butterfly operations are independent of each other only requiring the previous computations to have been completed, a single hardware unit can be used to perform these calculations for different input pairs in a multicyle operation provided that the intermediate results from each stage are stored. Consequently, the total number of cycles required to get the final result will be $Nn/2$ if each butterfly operation is performed in one clock cycle.

Fig. 5 shows a block diagram of the FFT processor in the sleep staging SoC. Data samples from the *input controller* are first read and ordered in a bit reversal pattern such that the new position of each sample is determined by flipping the binary bits of its original index. This also ensures the correct order of output samples at the end of the FFT computation. A set of register banks is used to hold the real and imaginary parts of the initial and intermediate complex values. The address generator keeps track of how many cycles and levels of computations have been performed. Based on the current cycle and level count, it generates addresses for the register banks to fetch and save data for each butterfly computation. The twiddle generator fetches the twiddle factor value from a lookup table for each computation based on the FFT cycle and level. By making use of the symmetry and periodicity properties to extract the twiddle factors, a total of only 64 complex values

need to be stored for a 512-point FFT. The butterfly operation is performed each cycle using the fetched twiddle factors and the data points. Its result is saved in place in the register bank and the fetch–compute–save operation is repeated in the next cycle. When the last of these calculations is complete, the register banks hold the resultant FFT coefficients. The magnitude for each complex output between 0.5 and 50 Hz is also calculated when the result at that index is available and stored in a separate register bank.

A counter within the FFT processor keeps track of the number of subepochs being processed. This value is passed on to the modules downstream and is used as the subepoch address. A *busy* status signal is raised high while the FFT computation is taking place. Once complete, this is set to low, which indicates to the input controller that the FFT process is free to start the next cycle of computation if valid data are available at its input. At the same time, the *flag_fft* signal is set to indicate that the FFT data are available for feature calculation.

2) *Power Calculation*: The power calculation module calculates the relative spectral power in different frequency bands using the magnitude values computed by the FFT processor. It normally stays in the idle mode and begins computation only after receiving the *flag_fft* signal from the previous module. The implementation of this module is shown in Fig. 6 and its operation involves summing up a set of input magnitude values in a multicyle operation. In the first cycle, a register is loaded with the first input from the *data_fft* input bus and an internal counter is incremented. In the subsequent cycles, other inputs are accumulated to this registered value until the counter reaches the maximum (equal to the total number of inputs). The value in the register at the end is then the power value for the current subepoch and a flag is raised to signify

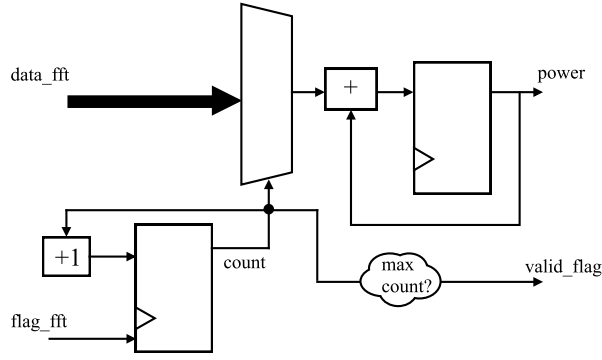


Fig. 6. Block diagram of the power calculation module.

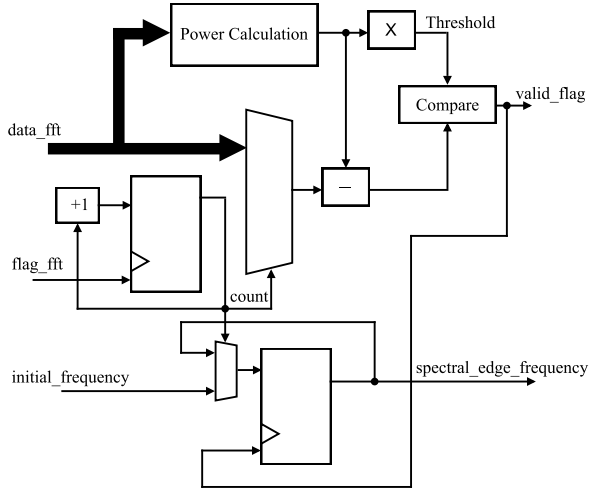


Fig. 7. Block diagram of the SEF calculation module.

that it is ready to be read. Several instances of this module are used with different FFT magnitudes as input corresponding to the frequency bands in which the power values are to be calculated.

3) *Spectral Edge Frequency Calculation*: The spectral edge frequency (SEF) calculation module is used to calculate the SEF values at 50% and 95% (*SEF50* and *SEF95*) in various frequency bands. It works by comparing the sum of powers in a frequency range against a given threshold. The threshold itself is obtained by multiplying the total power in a given frequency range by the desired percentage (edge). From the hardware point of view, both *SEF50* and *SEF95* modules are the same with one difference. For *SEF50*, multiplication by 0.5 is achieved by shifting 1 b to the right. In case of *SEF95*, multiplication by 0.95 is performed using an unsigned fixed-point multiplier with one input set to this constant value.

As shown in Fig. 7, at the start of each SEF calculation, the power in a given frequency range is obtained using the power calculation module (described in the previous section). This is multiplied by either 0.5 or 0.95 to establish the threshold. At the same time, the maximum frequency value in the given range is also registered. A multicycle operation then follows in which the power at the current highest frequency bin is

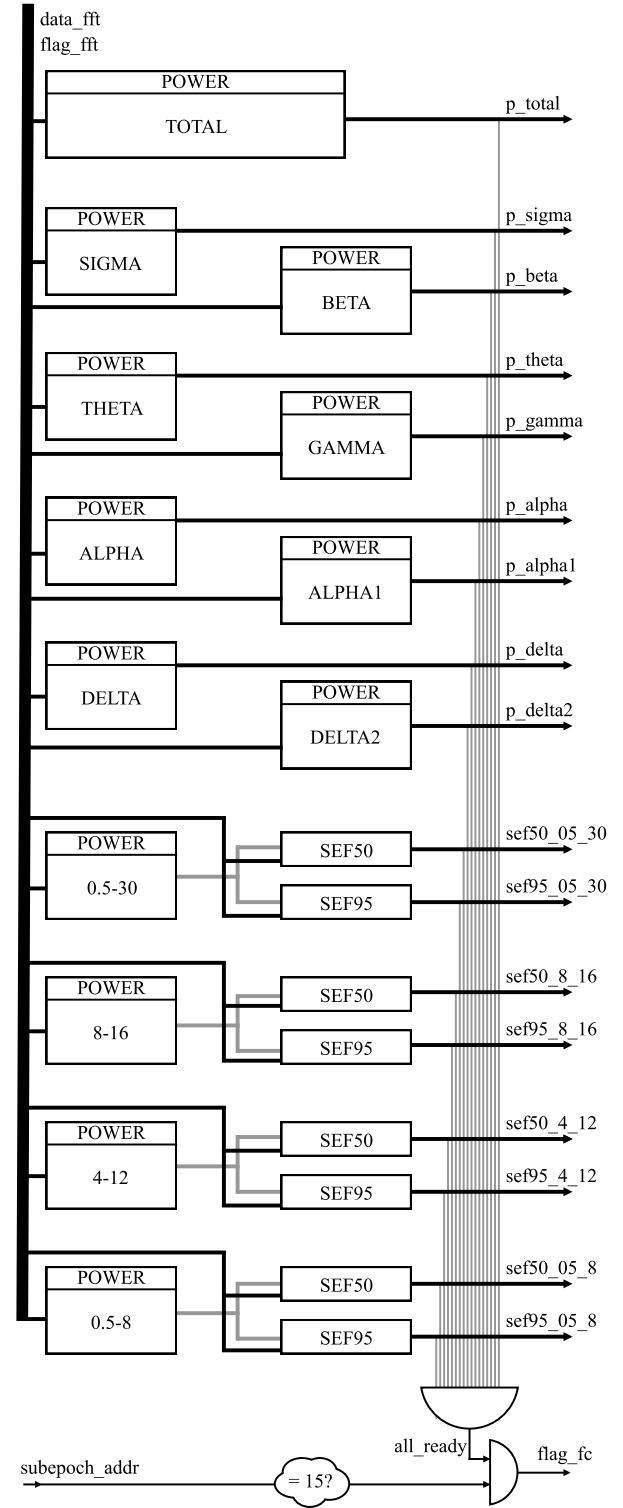


Fig. 8. Top level diagram of the feature calculation block.

subtracted from the total power. The maximum frequency value is also reduced by 0.5 (frequency resolution) each cycle and updated in the register. When the subtracted power value falls below the established threshold, the corresponding frequency value in the register is the required SEF.

4) *Block Level Implementation of Feature Extractor*: Fig. 8 shows the complete *feature extractor* block with multiple

TABLE I
DIFFERENT STATES OF THE FSM CORRESPONDING
TO THE CORE AND PERIPHERAL TESTS

Core Tests	Peripheral Tests
CORE_W	PERI_W_N1
CORE_N1	PERI_W_N2
CORE_N2	PERI_W_N3
CORE_N3	PERI_W_R
CORE_R	PERI_N1_N2
	PERI_N1_N3
	PERI_N1_R
	PERI_N2_N3
	PERI_N2_R
	PERI_N3_R

instances of *SEF50*, *SEF95*, and power calculation modules in different configurations to cover the various frequency ranges for all the features. When all the features for an epoch have been calculated, the `flag_fc` status signal is raised to indicate that they are ready to be used for classification.

D. Classifier

The *classifier* is the final stage within the algorithm pipeline that assigns a valid sleep stage to each 30-s epoch of the input EEG signal. It is an implementation of the finite state machine (FSM) approach described in Section IV. It has different states, as shown in Table I, with each state corresponding to a core or peripheral test. In addition, it also has an *IDLE* state during which it waits for valid features to be presented at its input.

When the SoC is powered on, the classifier is in the *IDLE* state. Its next state depends on the score of the last classified epoch and can be one of *CORE_W*, *CORE_N1*, *CORE_N2*, *CORE_N3*, or *CORE_R*—each corresponding to one of the core decision trees. The enable signal for the core test corresponding to this new state is asserted low to switch it ON from an idle state. The FSM then remains in the same state until all the computations of the core test are complete. If the core test classifies the epoch having the same sleep stage as the previous epoch, then the FSM goes back to the initial *IDLE* state since there are no further tests required. If the sleep stage is other than the current one, then the next state of the FSM is one of the peripheral tests. The peripheral tests are also enabled only when needed and their result determines the next state of the FSM until the epoch is assigned a sleep stage.

All core and peripheral tests are similar in design since they take a fixed number of features as inputs, perform one or more comparisons, and produce one of the sleep stages as output. The differences between them include the number of comparisons needed, the input features, and the thresholds corresponding to each feature. This in turn dictates how many clock cycles are required by each test to produce the result. Each node in the decision tree performs a comparison of only one feature against its specific threshold. The feature at the node can be one of the following three types.

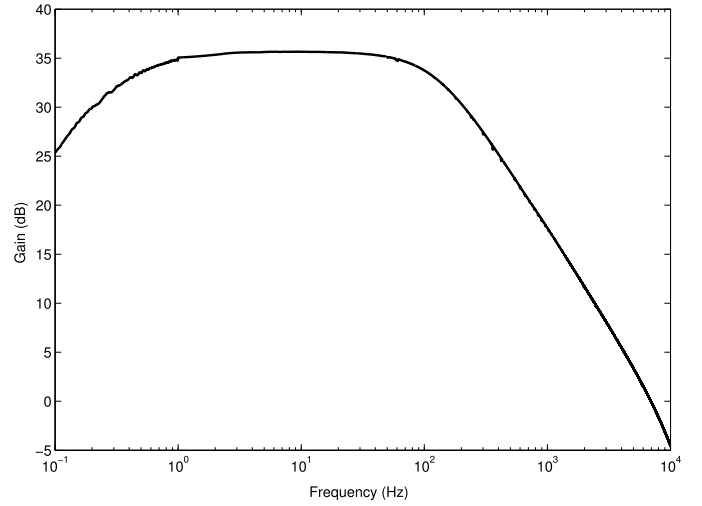


Fig. 9. Measured gain-transfer function of the amplifier.

- 1) *Relative Power in a Frequency Band*: This is the power in a certain frequency band divided by the power in the entire frequency of interest.
- 2) *Ratio of Power in Different Frequency Bands*: This is the power in a certain frequency band divided by the power in a different frequency band.
- 3) *Mean SEF*: This is obtained by dividing *SEF* by 15 (the number of subepochs in a 30-s epoch).

The three feature types are obtained in a similar fashion by dividing two numbers and then comparing this result against a threshold. This operation can be implemented using a single multiplier/divider and a comparator. At any given time, only one of the core or peripheral test is enabled. Within this test, in one clock cycle, only one multiplier and comparator are required across *all* core and peripheral tests since only one such comparison takes place in a single clock cycle. As a result, only one of each component can be shared across the different tests by multiplexing the comparator and multiplier inputs based on which test is currently enabled.

Fig. 10 shows the block diagram of the *classifier* with instances of all core and peripheral tests. Each test is provided with a subset of features that are needed for decision making and an enable signal that activates the test. The tests are connected to a multiplexer that controls the inputs to a single multiplier and a comparator. A state machine controller determines which test needs to be activated at any time, establishes the state of the machine, and also provides the control signal for the multiplexer to allow correct inputs for the shared datapath. The final result of classification is available on the 3-b output port. It is encoded as shown in Table II and remains valid until the next epoch is available at the classifier input.

VI. SOC TEST RESULTS

A. Measurements

The sleep staging SoC is implemented using the AMS 0.18 μm process technology with six metal layers. The total chip area includes the analog frontend 10.3 mm^2 . The digital blocks measure 8.89 mm^2 in die area with a logic gate count of about 127k. The gain-transfer function of the amplifier in Fig. 9 shows the gain to be 36 dB in the frequency range of

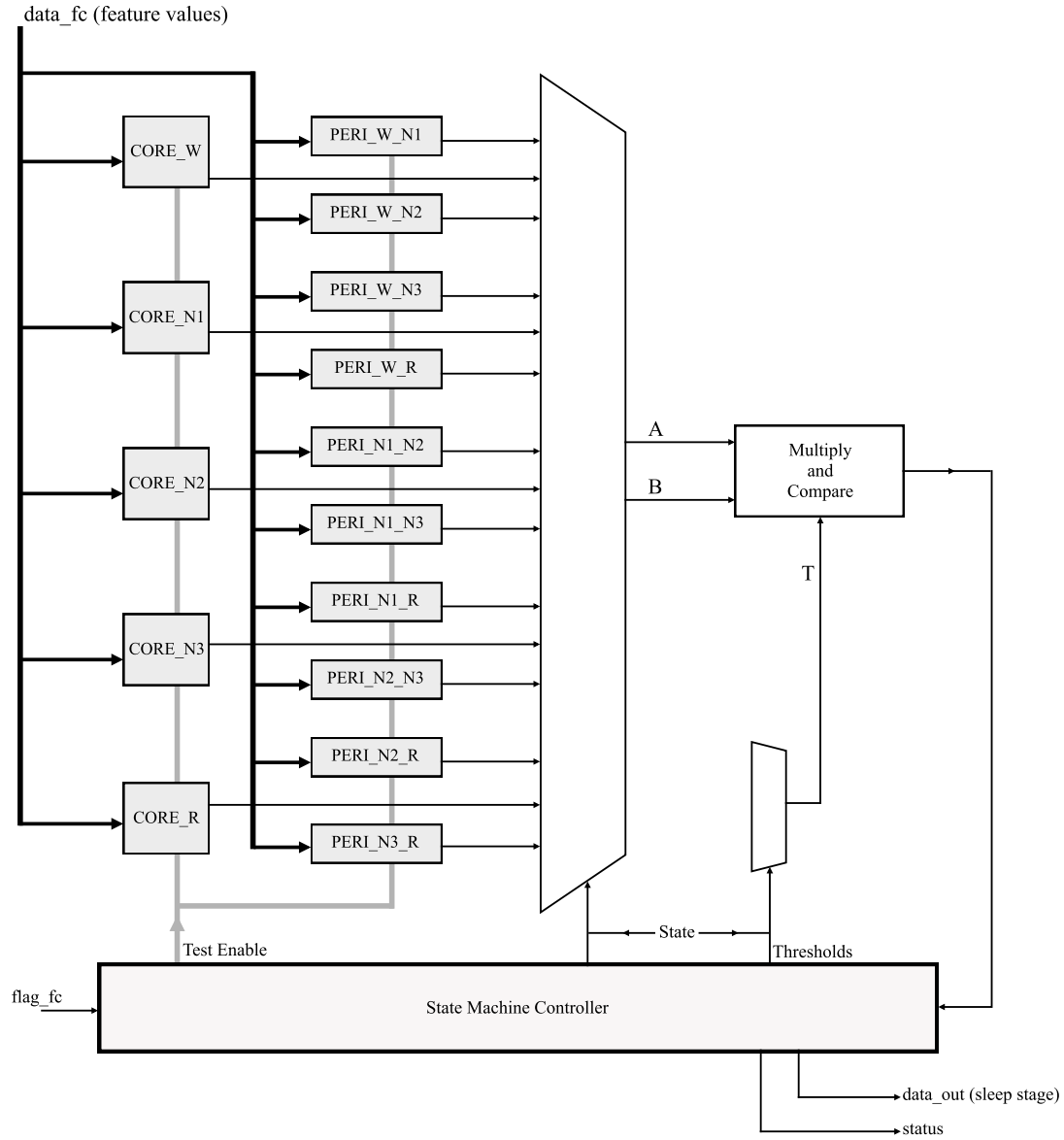


Fig. 10. Top level diagram of the classifier block.

TABLE II
ENCODING OF THE OUTPUT FROM THE CLASSIFIER BLOCK

Sleep Stage	Output Encoding
Wake	000
N1	001
N2	010
N3	011
REM	100
Others/Unknown	111

interest. The SoC powered using a single 1.25 V supply and its average power consumption is 575 μ W, while operating at a clock frequency of 1.5 kHz. Its performance summary is listed in Table III and a die photograph of the IC is shown in Fig. 11.

The power and area breakdowns of the complete SoC are shown in Fig. 12. It can be seen that the digital processing

blocks consume about three quarters of the total chip power. They have been synthesized using Cadence RTL Compiler (version 11.21) [18] with several options to reduce their power consumption. These include automatic insertion of clock gates using slow speed datapath components since the SoC is required to run at a low clock frequency, multithreshold voltage optimization, and insertion of operand isolation. Apart from the tool-aided reduction in power, other methods were also implemented to achieve power savings. Since both core and peripheral tests are seldom activated, the clocks to these tests are gated and enabled when classification using a particular test is needed. Similarly, the clock input to the FFT processor is also gated and switched off when the input controller is collecting data. Further, the algorithm itself was developed with the intent of hardware implementation and allows using a small number of computations at the classification stage in order to reach a decision. These design techniques are not exhaustive and other methods such as power

TABLE III
SoC PERFORMANCE SUMMARY

Technology	AMS 0.18 μm CMOS (C18A6)
Power Supply	1.25 V
ADC Resolution	12 bits
Gain (dB)	36
CMRR (dB)	62
Logic Gate Count	127K
Total Chip Area	10.3 mm ²
Digital Clock Frequency	1.5 kHz
Analog Clock Frequency	8.192 kHz
Sampling Rate	256 Hz
Classification Accuracy	98.72%
Total Power Consumption	575 μW
Analog Front-End	149 μW
Digital Processor	426 μW

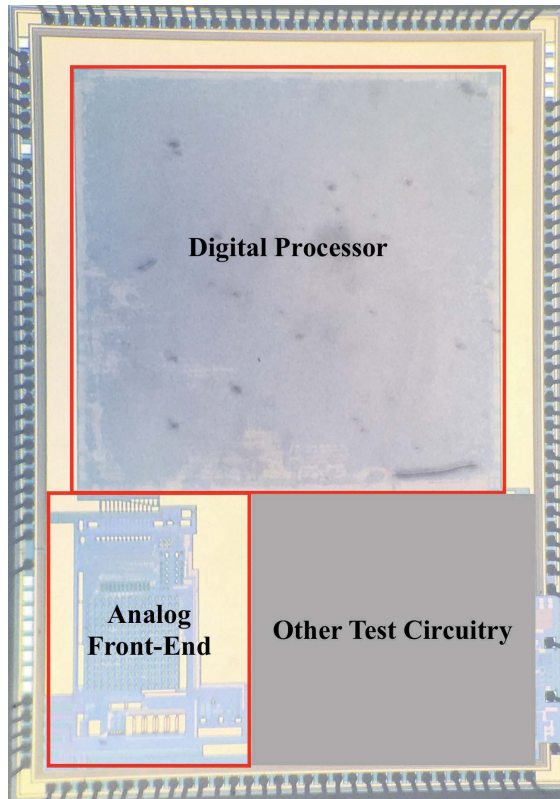


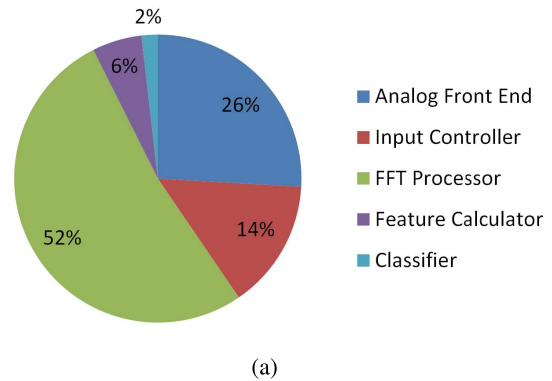
Fig. 11. Picture of the fabricated sleep staging SoC showing the analog frontend and digital processor.

gating, multiple supply voltages to lower voltage for certain blocks, and algorithmic improvements can be effectively used to further reduce the power consumption. Additionally, the FFT processor, which consumes about half of the overall SoC power can be improved by following the implementations in [19] and [20] and by exploring more efficient architectures [21], [22].

B. System Demonstration

The performance of the sleep staging SoC was verified using real EEG input while measuring the chip output at

Power Breakdown



Area Breakdown

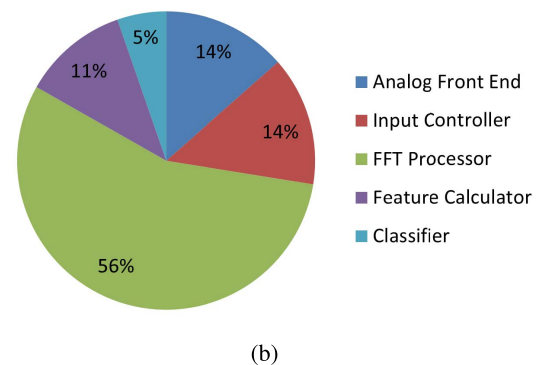


Fig. 12. (a) Power and (b) area breakdowns of the sleep staging SoC.

intermediate and final stages to ensure operational integrity of each block. Fig. 13 demonstrates this with a 90-s input EEG signal (equivalent to three 30-s epochs). The algorithm, however, processes data in blocks of 2 s, computing the FFT for each block and calculating different features from this result. The acquired data from a subepoch of the first 2 s, followed by a plot its FFT spectrum, as calculated on the chip,

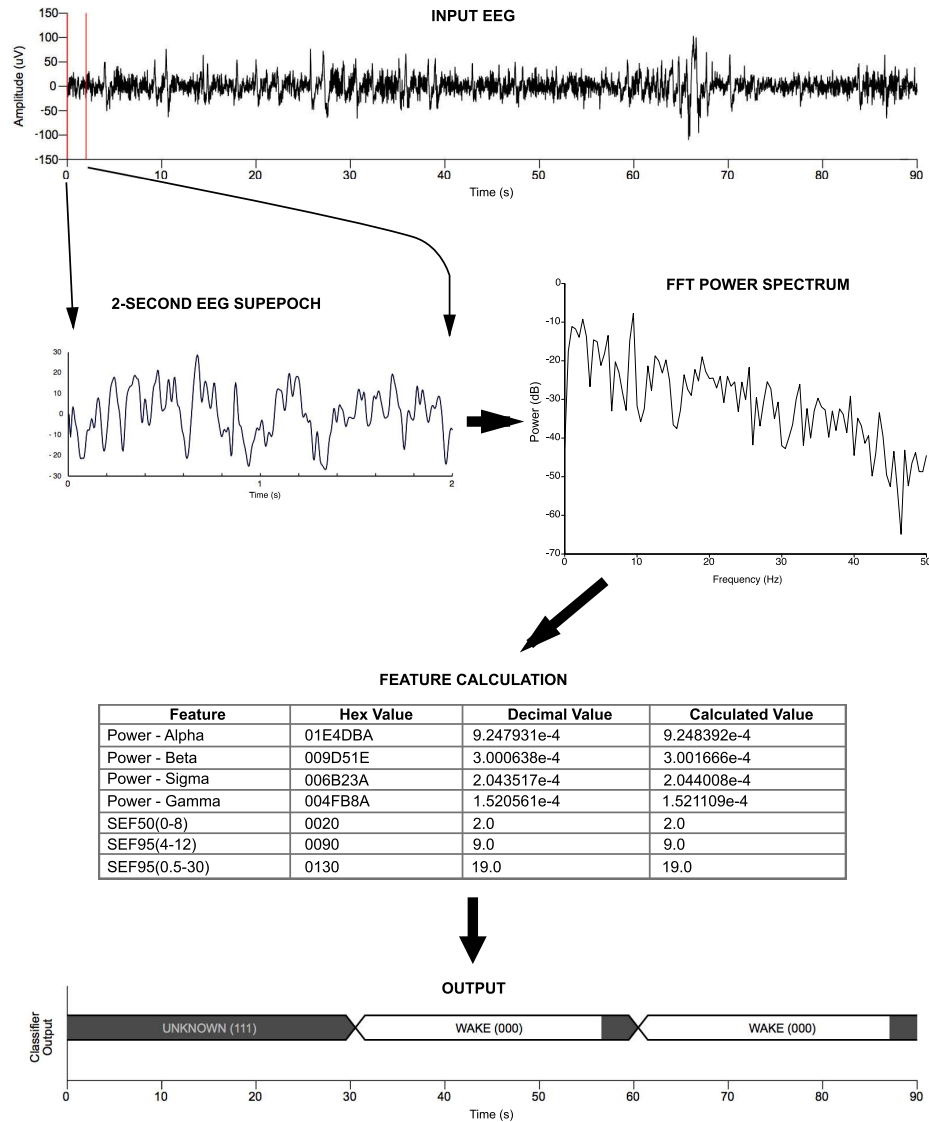


Fig. 13. Demonstration of the sleep staging SoC using real EEG signal with intermediate and final outputs.

is shown in Fig. 13. Using these FFT magnitudes, the different feature values are calculated. A subset of these feature values, together with their MATLAB-calculated equivalents, is also shown in Fig. 13 for comparison. Finally, the last plot shows the 90-s output, corresponding to the input EEG. On this plot, the output is invalid during the initial 30 s. After this time, the classifier output for the first epoch is available and encoded as 000 (Wake). It becomes invalid briefly before the output for the second epoch appears. This is indicated by a 1-b signal that goes low whenever there are invalid data on the output port.

C. Classification Accuracy

The classification accuracy of the sleep staging SoC is determined by comparing its performance with that of the algorithm that has been developed in MATLAB. The reference algorithm itself used overnight EEG recordings (Fp1-A2 channel) from 20 human subjects in the DREAMS Subjects

database [23]. Using the same real EEG signals as input, the overall classification accuracy of the SoC is 98.72% compared with that of the reference algorithm. Ideally, this should be 100%, but there are a number of factors that make the hardware different from the reference algorithm. The main difference, however, is that the reference algorithm uses 64-b floating-point number representation, while the hardware has been designed using 24-b fixed-point format for representing each number. It is possible to use 64-b floating-point numbers on the SoC as well, but the increase in power consumption and area requirements can be up to four times compared with the fixed-point implementation [24]. Hence, while the use of fixed-point numbers adds truncation and round-off errors leading to misclassified epochs, the reduction in accuracy is very small compared with the additional resources that would be needed for the floating-point implementation. Further, it must be noted that in most cases, a misclassified epoch does not cost the algorithm significantly in terms of the overall accuracy. Once

TABLE IV
COMPARISON OF SoC PERFORMANCE WITH OTHER RELATED SYSTEMS

	Verma [27] JSSC 2010	Yoo [28] JSSC 2013	Lee [29] JSSC 2013	Chen [30] JSSC 2014	Altaf [31] JSSC 2015	This Work
Signal	EEG	EEG	EEG	iEEG	EEG	EEG
Application	Seizure	Seizure	Seizure	Seizure	Seizure	Sleep
# Channels	1	8	1	8	16	1
Feature Extraction	o	o	o	o	o	o
On-chip Classification	x	o	o	o	o	o
AFE Noise RTI	1.3 μ V (0-100Hz)	0.9 μ V (0.5-100Hz)	-	5.23 μ V	0.9 μ V (0.5-100Hz)	2.52 μ V (0.3-130Hz)
Input impedance	> 700 M Ω	> 500 M Ω	-	-	> 500 M Ω	> 500 M Ω
Gain (dB)	40	40	-	41-61	40	36
CMRR (dB)	60	-	-	-	97	62
IA NEF	-	5.12	-	1.77	3.29	5.7
Supply (V)	1.0	1.8/1.0	0.55-1.2	1.8	1.8/1.0	1.25
Classifier	x	Linear SVM	SVM	LLS	D2A-LSVM	Decision Trees
Energy	9 μ J/feature	2.03 μ J/class	273 μ J/class	77.9 μ J/class	2.73 μ J/class	0.7 μ J/class
Technology	0.18 μ m	0.18 μ m	0.13 μ m	0.18 μ m	0.18 μ m	0.18 μ m

an epoch is incorrectly classified, the state machine controlling the decision trees goes into the wrong state. However, it is expected that the next set of core and peripheral tests will bring it back to the correct state.

VII. DISCUSSION

This paper discusses the design of a complete sleep staging SoC using a single EEG channel. To the best of authors' knowledge, this is also the first implementation of a complete sleep staging algorithm on chip. As a result, it is difficult to make direct comparisons of its performance with those of other systems that have been designed for different applications. Instead, a performance comparison table of some common circuit blocks that have been used in related state-of-the-art systems is shown in Table IV. However, it should be noted that these systems have been designed for entirely different applications. In spite of it, a number of conclusions can be drawn from the work presented in this paper in the context of wearable technologies. The most important conclusion is that having a customized on-chip sleep staging system can result in such low power levels that a one channel EEG system could be running for 370 h on a typical hearing-aid battery [25]. Considering also that the size of the electronics would also be reduced because of the SoC implementation and that a typical off, the shelf electrode is 10 mm in diameter [26] and the overall size of the system can be extremely small. This illustrates how having custom implementation of algorithms on chip can significantly optimize the usability related specifications for wearable technologies, and specifically within the context of this work for sleep monitoring technologies.

Although it could be argued that there are limitations when using just one EEG channel—such as a higher risk of completely losing the signal (associated with the electrode being deattached) and a reduction in accuracy due to not having EMG or EOG electrodes, the advantages of having just one channel outweighs the downsides, mostly in the context of certain applications. For example, even when a one-channel system may not be a complete replacement for full PSG carried out in a specialist sleep clinic, it is certainly much more accurate and detailed than actigraphy and can serve as an initial at-home screening device. Hence, using just one channel not only reduces the volume of the system as a whole but also increases its usability (by making it easier to attach). In addition, outside the context of diagnosis, having a long-lasting and easy-to-use wearable sleep staging system can allow for the possibility of large clinical trials for sleep medicine, which would not be possible with traditional PSG systems.

VIII. CONCLUSION

The sleep staging SoC presented here uses an analog frontend for sensing data and a digital processor to run an automatic sleep staging algorithm that has been specifically developed to suit low-power and resource-constrained systems. Its implementation uses a small number of datapath components for feature extraction and classification. Further, all blocks within the system remain in an idle mode waking up only when needed. As a result of its on-chip implementation, the power consumption and area requirements are greatly

reduced making the final package small and suitable for wearable use. This demonstrates the potential of our work to be used either as part of a stand-alone sleep monitoring system or as part of a broad-based multipurpose EEG system on chip that is not only limited to sleep monitoring but also useful for other neurological conditions such as epilepsy, stroke, or Alzheimers.

REFERENCES

- [1] C. Iber, S. Ancoli-Israel, A. Chesson, and S. Quan, *The AASM Manual for the Scoring of Sleep and Associated Events: Rules, Terminology and Technical Specifications*. Westchester, IL, USA: American Academy Sleep Medicine, 2007.
- [2] M. Partinen and C. Hublin, "Epidemiology of sleep disorders," in *Principles and Practice of Sleep Medicine*, M. H. Kryger, T. Roth, and W. C. Dement, Eds. Amsterdam, The Netherlands: Elsevier, 2011, ch. 61.
- [3] H. K. Yaggi, J. Concato, W. N. Kernan, J. H. Lichtman, L. Brass, and V. Mohsenin, "Obstructive sleep apnea as a risk factor for stroke and death," *New England J. Med.*, vol. 353, no. 19, pp. 2034–2041, 2005.
- [4] J. A. Horne and L. A. Reyner, "Sleep related vehicle accidents," *Brit. Med. J.*, vol. 310, no. 6979, pp. 565–567, 1995.
- [5] G. Stores, "Clinical diagnosis and misdiagnosis of sleep disorders," *J. Neurol., Neurosurgery Psychiatry*, vol. 78, no. 12, pp. 1293–1297, 2007.
- [6] J. L. Martin and A. D. Hakim, "Wrist actigraphy," *Chest*, vol. 139, no. 6, pp. 1514–1527, 2011.
- [7] M. Ronzhina, O. Janoušek, J. Kolářova, M. Nováková, P. Honzík, and I. Provazník, "Sleep scoring using artificial neural networks," *Sleep Med. Rev.*, vol. 16, no. 3, pp. 251–263, 2012.
- [8] H. Danker-Hopfe *et al.*, "Interrater reliability between scorers from eight European sleep laboratories in subjects with different sleep disorders," *J. Sleep Res.*, vol. 13, no. 1, pp. 9–63, 2004.
- [9] H. Danker-Hopfe *et al.*, "Interrater reliability for sleep scoring according to the Rechtschaffen & Kales and the new AASM standard," *J. Sleep Res.*, vol. 18, no. 1, pp. 74–84, 2009.
- [10] W. R. Ruehland *et al.*, "The 2007 AASM recommendations for EEG electrode placement in polysomnography: Impact on sleep and cortical arousal scoring," *Sleep*, vol. 34, no. 1, pp. 73–81, 2011.
- [11] S. Imtiaz and E. Rodriguez-Villegas, "Automatic sleep staging using state machine-controlled decision trees," in *Proc. IEEE EMBC*, Milan, Italy, Aug. 2015, pp. 378–381.
- [12] Bluetooth Technology Website. (2017). *Bluetooth Low Energy*. [Online]. Available: <https://www.bluetooth.com/what-is-bluetooth-technology/how-it-works/low-energy>
- [13] J. R. Smith, M. J. Cronin, and I. Karacan, "A multichannel hybrid system for rapid eye movement detection (REM detection)," *Comput. Biomed. Res.*, vol. 4, no. 3, pp. 275–290, 1971.
- [14] J. C. Principe and J. R. Smith, "SAMICOS—A sleep analyzing microcomputer system," *IEEE Trans. Biomed. Eng.*, vol. 33, no. 10, pp. 935–941, Oct. 1986.
- [15] P. E. Allen and D. R. Holdberg, *CMOS Analog Circuit Design*. Oxford, U.K.: Oxford Univ. Press, 2002.
- [16] R. R. Harrison and C. Charles, "A low-power low-noise CMOS amplifier for neural recording applications," *IEEE J. Solid-State Circuits*, vol. 38, no. 6, pp. 958–965, Jun. 2003.
- [17] J. W. Cooley and J. W. Tukey, "An algorithm for the machine calculation of complex Fourier series," *Math. Comput.*, vol. 19, no. 90, pp. 297–301, 1965.
- [18] Cadence. (2015). *Encounter RTL Compiler*. [Online]. Available: http://www.cadence.com/products/ld/rtl_compiler/pages/default.aspx
- [19] S. R. Sridhara *et al.*, "Microwatt embedded processor platform for medical system-on-chip applications," *IEEE J. Solid-State Circuits*, vol. 46, no. 4, pp. 721–730, Apr. 2011.
- [20] J. Kwong and A. P. Chandrakasan, "An energy-efficient biomedical signal processing platform," *IEEE J. Solid-State Circuits*, vol. 46, no. 7, pp. 1742–1753, Jul. 2011.
- [21] Z. Qian and M. Margala, "Low-power split-radix FFT processors using radix-2 butterfly units," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 24, no. 9, pp. 3008–3012, Sep. 2016.
- [22] M. Garrido, "A new representation of FFT algorithms using triangular matrices," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 63, no. 10, pp. 1737–1745, Nov. 2016.
- [23] University of MONS—TCTS Laboratory. (2015). *The DREAMS Subjects Database*. [Online]. Available: <http://www.tcts.fpm.ac.be/~devuyt/Databases/DatabaseSubjects/>
- [24] A. Rushton, *VHDL for Logic Synthesis*, 3rd ed. Chichester, U.K.: Wiley, 2011, ch. 15, pp. 337–367.
- [25] VARTA Microbattery GmbH. (2016). *Zinc Air Wireless Approved Mercury-Free P312*. [Online]. Available: <http://www.powerone-batteries.com/en/products/zinc-air-mercury-free-batteries/p312/>
- [26] Unimed Electrode Supplies. (2016). *EEG Electrodes*. [Online]. Available: <http://www.unimed-electrodes.co.uk/eeeg>
- [27] N. Verma, A. Shueb, J. Bohorquez, J. Dawson, J. Gutttag, and A. P. Chandrakasan, "A micro-power EEG acquisition SoC with integrated feature extraction processor for a chronic seizure detection system," *IEEE J. Solid-State Circuits*, vol. 45, no. 4, pp. 804–816, Apr. 2010.
- [28] J. Yoo, L. Yan, D. El-Damak, M. A. B. Altaf, A. H. Shueb, and A. P. Chandrakasan, "An 8-channel scalable EEG acquisition SoC with patient-specific seizure classification and recording processor," *IEEE J. Solid-State Circuits*, vol. 48, no. 1, pp. 214–228, Jan. 2013.
- [29] K. H. Lee and N. Verma, "A low-power processor with configurable embedded machine-learning accelerators for high-order and adaptive analysis of medical-sensor signals," *IEEE J. Solid-State Circuits*, vol. 48, no. 7, pp. 1625–1637, Jul. 2013.
- [30] W. M. Chen *et al.*, "A fully integrated 8-channel closed-loop neural-prosthetic CMOS SoC for real-time epileptic seizure control," *IEEE J. Solid-State Circuits*, vol. 49, no. 1, pp. 232–247, Jan. 2014.
- [31] M. A. B. Altaf, C. Zhang, and J. Yoo, "A 16-channel patient-specific seizure onset and termination detection SoC with impedance-adaptive transcranial electrical stimulator," *IEEE J. Solid-State Circuits*, vol. 50, no. 11, pp. 2728–2740, Nov. 2015.



Syed Anas Imtiaz (S'07–M'16) received the B.Eng. degree from the National University of Sciences and Technology, Islamabad, Pakistan, in 2008, and the M.Sc. and Ph.D. degrees from Imperial College London, London, U.K., in 2009 and 2015, respectively.

From 2009 to 2010, he was a Digital Design Engineer at Imagination Technologies, Kings Langley, Hertfordshire, U.K. He is currently focusing on creating novel wearable technologies to aid long-term monitoring and diagnosis of different medical conditions. His current research interests include developing low-complexity signal processing algorithms and their low-power mixed-signal circuit design, particularly for use in sleep medicine and epilepsy monitoring.



Zhou Jiang received the B.Eng. degree in electrical and electronic engineering from the University of Bristol, Bristol, U.K., in 2011, and the M.Sc. degree in integrated circuit design from Imperial College London, London, U.K., in 2012, where he is currently pursuing the Ph.D. degree with the Circuits and Systems Group.

His current research interests include analog and digital integrated circuit designs for miniature wireless neuronal activity recording systems, ultralow power analog, and RF integrated circuit designs for

wireless applications.



Esther Rodriguez-Villegas (SM'08) is currently a Full Professor in Low Power Electronics with the Department of Electrical and Electronic Engineering, Imperial College London, London, U.K., where she is involved in ultralow-power electronic circuits and systems for truly wearable physiological monitoring. She has authored more than 100 peer-reviewed papers and a book on *FGMOS Transistors* (IET, 2006).

Prof. Rodriguez-Villegas is a member of the Technical Committees in several international conferences such as the IEEE International Symposium on Circuits and Systems, the IEEE International Conference on Electronics, Circuits and Systems, and the IEEE Biomedical Circuits and Systems. Her research has led to numerous awards including the Nokia Sensing XPrize in 2014, the IET Innovation Award, and the Complutense Young Award for Science and Technology in 2009. In 2010 she organized a mini-symposium on truly wearable medical devices at the IEEE Engineering in Medicine and Biology Conference.