9. Towards A MASC Design

9.1 Overview

In the last chapter, the notion of an Oscillator Descriptor (OD) is developed which offers an object-oriented insight into MASC execution. Also, the architectural context of a single external data-bus MASC accessing a single port Shared Memory (SM) under mutual exclusion with a Host CPU is developed, based upon the classical principle of memory hierarchy. Combination of these complementary principles leads to the idea of control data bandwidth optimisation in the MASC-SM data-bus via interleaved burst processing as summarised in section 8.5.2. These proposals are low-level solutions to the problem of handling high AS control bandwidth in a hardware implementation (as identified in section 1.1.2) and there remains the task, resolved in this chapter, of specifying the additional higher-level functionality for a complete MASC design.

First, the theme of section 8.5.2 is developed for a MAS context. The resulting ‘frame-based’ timing structure leads to scheduling, resource allocation and synchronisation solutions for realising the logical process pipeline of section 8.3.1. Much of the MASC functional specification is based upon observations of the relative control bandwidth of the constituent processes, illustrated qualitatively in Fig. 8.2. For a more quantitative estimate of MASC efficiency, the assignment of practical values to MASC design parameters leads to an understanding of how computational optimisation via MAS is manifest in the MASC, and of which design parameters have the largest impact on MASC performance. Finally, the dataflows and low-level logic diagrams for the major functional units within the MASC are documented.

9.2 MASC Functional Requirements for MAS

9.2.1 Supporting Multiple Sample Rates for MAS by Frame Scheduling

In the MOB burst processing scheme of section 8.5.2, the OD schedule period increases from $1/f_s$, as required by a TOB, to N_{burst}/ f_s - where N_{burst} is the length of an MOB burst - in order that oscillators are updated at an aggregate rate of f_s. This longer schedule is designated as a ‘frame’ and is the highest level timing construct in the MASC architecture, in contrast to the MOB burst, which is the lowest. Frames facilitate a simple and efficient transformation of the single sample rate MASC assumed in Chapter 8 into a multirate form for MAS. The frame period is set to T_{frame} as expressed in eqn (9.1) where the denominator is the sample rate of the lowest level of the subband hierarchy of depth *K*. Therefore the MASC is optimised for AS in the lowest subbands as illustrated in Fig. 8.5 where MASC-SM IPC is saturated. Higher aggregate sample rates for an oscillator *x* allocated to a subband at level k:($0 \le k < K$) are generated by *extending* the MOB burst to $2^{K-k} N_{burst}$, that is to say multiples of 8,4 and 2 of N_{burst} for k=0,1,2 in a hierachy of depth K=3. Fig. 9.1 shows the impact of an extension of burst length to $2N_{burst}$ upon the single sample rate scheme illustrated in Fig. 8.5. which generates the correct aggregate sample rate for subbands at level k=K-1 i.e. the next subband series above the base level at *K*.

$$T_{frame} = \frac{N_{burst}}{f_s 2^{-K}} \tag{9.1}$$

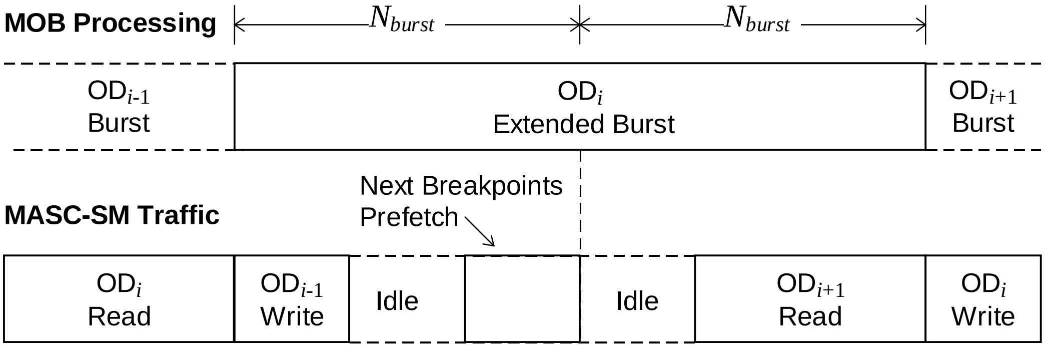

Figure 9.1 Extended Burst of $2N_{burst}$ for Subbands at Level k=K-1

Extended MOB bursts have the effect of introducing idle cycles into, otherwise, optimally interleaved SM-MASC traffic. However, to maintain a control rate proportional to subband sample rate in agreement with Heisenberg’s principle (by breakpoint quantisation to integer multiples of N_{burst}) successor $A_i[n]$ and $F_i[n]$ breakpoints are prefetched every N_{burst} sub-division of the extended burst in anticipation of a possible succession. Only four words are accessed in place of the twelve required for a complete OD update and therefore the SM is idle for most of the extended burst beyond the first N_{burst} sub-division which releases the SM for potential Host / DSP accesses without suspension of the MASC (a logical refinement would be to predict breakpoint succession so that breakpoint prefetch occurs only when necessary). A beneficial property of the frame is that it can support any distribution of MAS sample rates with the constraint that the net duration of all S bursts in the schedule must not exceed T_{frame}. Optimisation by MAS is reflected in the fact that extended bursts of higher subbands in the subband hierarchy ($k<K$) occupy a larger share of MOB throughput than non-extended bursts at the optimal level of K.

9.2.2 Burst Accumulation of Subband Streams

A concluding operation in classical AS is the accumulation of S sinusoids into a single output stream. In MAS, this operation is factorised into the accumulation of n_{sb} subband streams which are combined during filterbank processing into n_{fb} streams for effects processing: the MASC is responsible for the former operation. To ensure that accumulation introduces minimal roundoff noise, it is desirable that accumulation wordlength matches the output of the MOB $A_i[n]$ multiplier, which is the sum of the sine transform and $A_i[n]$ resolution, of the order of 16-bits apiece and therefore 32-bit accumulation is desirable. To integrate subband accumulation with interleaved burst processing as outlined in section 8.5.2 is straightforward if the OD schedule is coalesced such that all OD's associated with individual subband streams are executed consecutively in time; these groupings of OD's are termed 'minor runs'.

In place of the single register in the accumulator loop at the output of a TOB, the MOB has a FIFO of length N_{burst} so that arrival of the current MOB burst at the accumulator coincides with a recirculated burst comprising the sum of previous oscillators in the minor run. However, the accumulation burst must be initialised and, upon completion, communicated to the relevant subband and filterbank. To facilitate this, the OD header contains subband and filterbank addresses which permit detection by the MOB of a new minor run: signified by a change of subband address between succeeding OD's. When such an event occurs, the MOB blocks update of the new OD until the accumulation FIFO is flushed to buffers for reading by the Filterbank Processor (FBP) constituted by the DSP in the Type-I topology and Host in the Type-II: the best location for these buffers is in a designated portion of the SM. Note that the MOB itself has a latency due to the inclusion of the proposed pipelined linear interpolated LUT of section 7.4.3 and therefore accumulation burst recirculation lags slightly behind the OD context switches at the MOB input. These points are illustrated in Figure 9.2 (for non-extended bursts) where T_{MOB} denotes MOB pipeline latency.

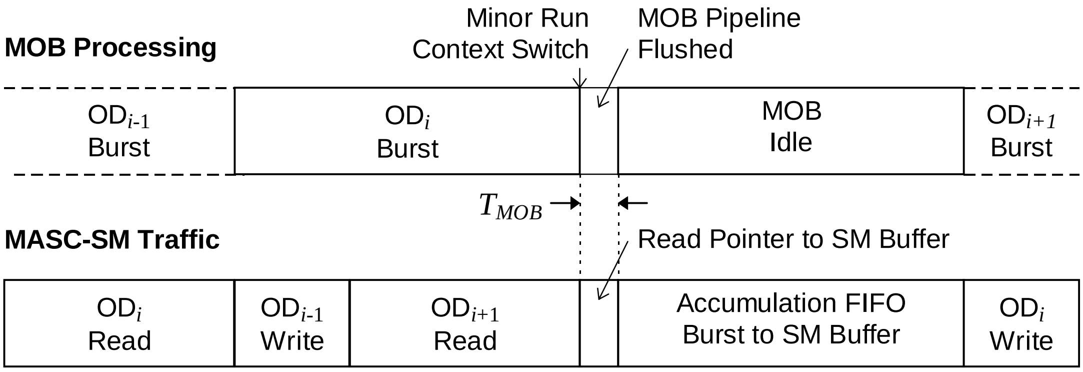

Figure 9.2 Accumulation Burst Timing

MOB blocking occurs upon completion of the first OD read of the new minor run because it coincides with the completion of the final MOB burst of the previous minor run where FIFO state is primed for flushing in order; a statement that is true for extended bursts also. Accumulation FIFO flushing starts T_{MOB} cycles after suspension of MASC-SM traffic because the internal MOB pipeline must be flushed first. It is proposed to exploit this small margin of SM redundancy by using the filterbank and subband address of the previous minor run (obtained from its last OD header) to access a pointer table in a reserved area of the SM which returns the absolute SM address of destination MOB-FBP buffers, permitting compaction of the latter by the Host in the presence of reconfigurable filterbanks in order to minimise SM fragmentation. This implies a conceptual ordering of the SM memory map by increasing memory requirements into (i) a fixed size MOB-FBP buffer pointer table indexed by subband and filterbank (ii) FBP-configured MOB-FBP buffers and (iii) the OD workspace.

Once accumulation FIFO flushing is complete, conventional MOB burst operation and interleaved MASC-SM traffic resumes immediately from the point where it was suspended. Accumulation of the new minor run is primed by routing the MOB burst of the first OD directly into the accumulation FIFO. To take into account extended bursts in MAS, the FIFO must support multiple accumulation burst lengths of $2^{K-k}N_{burst}$ for k:($0 \le k < K$). For K=3, this is facilitated by a cascade of four FIFO's in the accumulation loop of length 1,1,2 and $4 \times N_{burst}$, with multiplexers prior to the last three which permit them to be switched into, and out of, the accumulation loop. Bypass of none, the last one, the last two, through to bypass of all three provides, in decreasing order, the requisite FIFO sizes. All FIFO's may thus be implemented as simple delay-lines. The accumulation structure is shown in section 9.5.4.

9.3 Scheduling, Resource Allocation and Synchronisation

9.3.1 Data Structures and Algorithms for Scheduling and Resource Allocation

The requirement for coalescing OD's into minor runs is non-trivial when the needs of resource allocation in MAS are considered. Each minor run is likely to be in a state of flux as new oscillators are added at the onset of a note, and removed upon its completion. To express the execution order of OD's, the schedule for each frame itself must be represented as a higher-level data structure referencing OD's at a lower level in the SM. A 'first-cut' solution is to execute OD's in the SM each frame as a sequential table ordered on ascending SM address. Clearly, a problem is that for optimal processing on the lines of section 8.5.2, OD's must be contiguous in the SM with minimal fragmentation: an unallocated OD will lead to wasteful stalling of the MASC for an entire OD cycle. However, adding a new OD to a compacted table requires block memory displacement to free the space required, consuming an extravagant number of SM cycles. Therefore a level of indirection is required such that the schedule order is not related to the absolute SM address of OD's.

The alternative of a sequential table of pointers to OD's has the same problems as a table of OD's, only that the overheads of defragmentation are reduced because pointers rather than OD's are displaced. Therefore organising OD's into a linked list (or rather loop, as the schedule is 'round-robin') is the most efficient data structure for representing the schedule (Moorer, 1982). The pointer from the current OD to its successor is contained in the header word and is a truncated form of its absolute SM address (see section 9.5.3).

Therefore, scheduling information is loaded transparently by the MASC in the same number of SM cycles as proposed in section 8.4.2. It is advantageous to align OD's on boundaries less than maximum OD size (determined by EC scope) so that smaller OD's can be used where necessary to economise on SM space. For efficient resource management in the SM, it must be configured such that there is a schedule of allocated OD's, and also a single 'free-space' list of unallocated OD's ready for initialisation by the Host and splicing into the appropriate point of the schedule, using standard linked-list algorithms (Tenenbaum and Augenstein, 1986). Upon termination, OD's are deallocated by removal from the schedule and are then added to the head of the free-space list pending future reallocation. So long as mutual exclusion for manipulation of the schedule loop is obeyed (outlined in the next section), these operations encompass all of the resource allocation functionality required between the MASC and Host. For comparison, Limberis and Bryan (1993) provide a case study of dynamic resource allocation in a conventional synthesiser.

9.3.2 Frame-Level Process Synchronisation

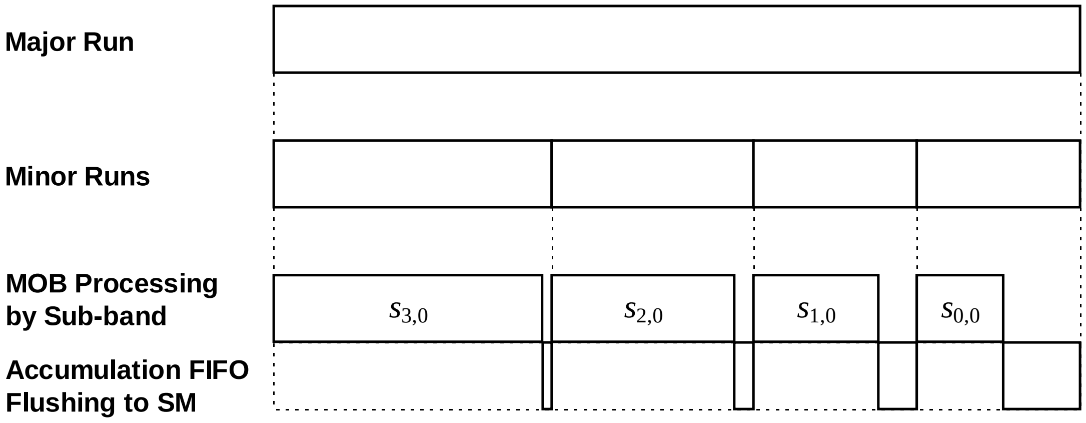

Figure 9.3 Minor / Major Runs in the OD Schedule

A linked-list schedule of OD's also facilitates correct synchronisation in (i) breakpoint IPC from the Host to the MOB and (ii) subband IPC from the MOB to the FBP in the presence of a dynamically modifiable schedule. The purpose of MASC synchronisation is to guarantee IPC between processes within an upper bound of T_{frame} which is chosen to be sufficiently long to permit a margin of short-time asynchronicity between processes in the logical pipeline by exploiting the latency tolerance T_{max}. As stated previously, the schedule is partially ordered by coalescing OD's into 'minor runs' that pertain to each subband. A logical next step is to introduce a higher-level ordering by coalescing minor runs into 'major runs' that pertain to each filterbank. The organisation of a major run is illustrated in Fig. 9.3, with an (arbitrary) example of four minor runs for each subband level k of a filterbank of depth K=3: the effect of accumulation burst extension is shown (not to scale). The order of minor runs in a major run is not extrinsically important, but it is convenient to order the whole schedule on three keys of (*filterbank#*, and subband k and l). In consequence, a housekeeping overhead is imposed upon the Host in maintaining schedule order, which is minimised by the dynamic storage methods discussed in the previous section.

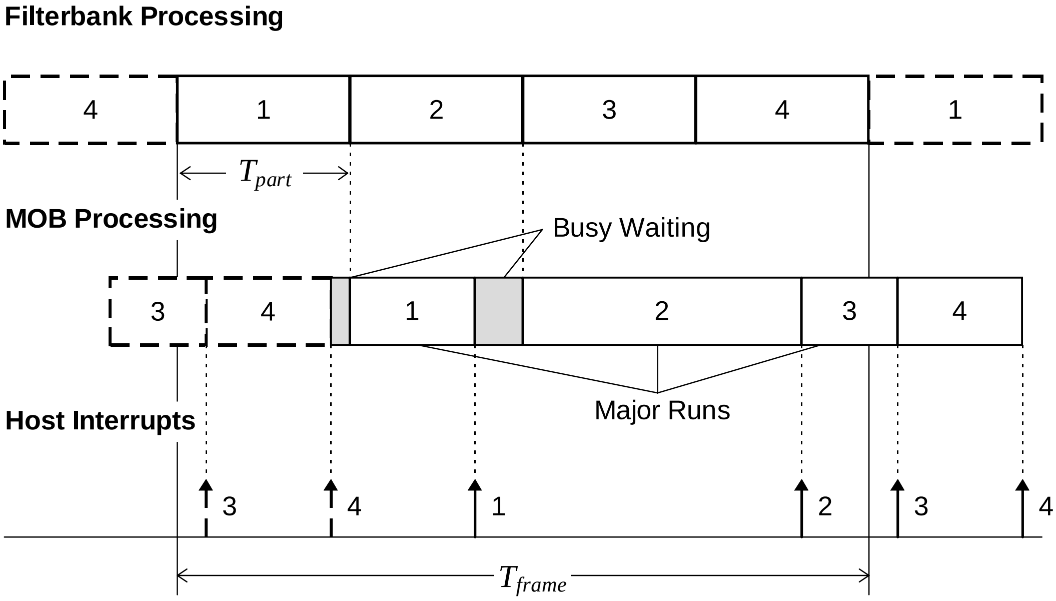

Figure 9.4 MASC Synchronisation for n_{fb}=4

The principal concept of the proposed frame-level synchronisation algorithm is that once ordered, MOB execution of the dynamic OD schedule is driven by the FBP which has, in contrast, a static schedule. Though the topology and number of filterbanks is reconfigurable, the configuration is assumed to be static during the 'run-time' of a MAS application. Fig. 9.4 illustrates an example for four filterbanks (n_{fb}=4) which can be extended to any value of n_{fb}. Each frame is divided equally up into n_{fb} partitions such that filterbanks are executed in ascending order from the start of the frame: each j^{th} partition denotes a time slot of duration T_{part} during which the FBP accesses only the SM buffers for filterbank j:$(1 \leq j \leq n_{fb})$. Filterbanks are executed depth-first by the FBP to generate a final burst of length $2^K N_{burst}$ which can be re-clocked via a FIFO at the required sample rate of f_s to the effects processor. Filterbank processing obeys principles of locality and may also take advantage of burst processing as far as FBP architecture permits. Filterbank topology is configurable implying a varying execution time but frame partitions are of equal duration. However, filterbank processing will occupy but a small margin of FBP throughput, and partitions are envisaged as being generated by a regular interrupt in the FBP system such that filterbank execution is completed within a modest interval at the start of each partition.

MOB execution of the major run for j must not overlap with the filterbank partition for j in order to maintain mutually exclusive access to SM buffers, otherwise there is a risk of data corruption. However, MOB operation can continue for other filterbanks. A simple algorithm to synchronise MOB processing to that of the FBP is (i) to order major runs in the OD schedule by ascending j and (ii) to allocate a synchronisation bit '*sync*' to each major run, located in a reserved position of the OD header and which only has significance in the first OD of a major run which is signified when the MOB detects a change in filterbank address in the header. If *sync*=1, filterbank j has completed its SM buffer access for the current frame and set *sync* thus so that major run j can be executed. However, if *sync*=0, filterbank j has not completed and the MOB enters a 'busy-waiting' state (Burns and Wellings, 1989) of repeatedly polling the header word (choice of an optimal polling frequency can release SM cycles for other potential operations). When the transition *sync*=0→1 is detected, major run j is initiated and the header of the first OD is written back with *sync*=0 to reset the synchronisation mechanism. Fig. 9.4 illustrates that contiguous execution of major runs is permitted when no SM buffer conflicts occur. Two constraints are imposed for ensuring the integrity of the proposed frame-level synchronisation algorithm in that (i) the maximum length of a major run is $(T_{frame}\text{-}T_{part})$ and (ii) the net duration of a set of consecutive major runs which contains no repeated index j must be less than T_{frame}. Note that margins must be included for expected FBP and Host SM access bandwidths. Under typical conditions in note-based music synthesis, these constraints are unlikely to be challenged, even in the presence of a dynamically configurable OD schedule that changes from frame to frame.

Host / MOB access to OD's under frame-level mutual exclusion is necessary during the initialisation and termination of oscillators. The MOB must execute only a completely initialised OD set for a new note so that oscillators commence execution correctly and partial envelopes are aligned in time. Also, the addition and removal of OD's requires temporary severing and manipulation of the schedule, violating its integrity and posing the danger that the MOB will reference invalid OD's, causing an MOB 'crash'. The mechanism for avoiding this state - also indicated in Fig. 9.4 - is for the MOB to send an interrupt signal to the Host upon completion of major run *j* indicating that there is an interval of time until the start of *j* in the next frame (a minimum of T_{part}) while *j* can be manipulated under mutual exclusion. At the end of the interval, schedule integrity must be restored by closing the reference loop. Frame-level conflicts are unlikely for OD update because the Host will interrogate the OD header for current status and supply breakpoints in advance of those currently queued. However, MOB interrupts provide the Host with a necessary mechanism for the real-time scheduling of OD updates. Contiguous MOB execution of a number of short major runs without 'busy-waiting' generates a burst of MOB interrupts which must be queued by the Host and processed sequentially. However, the response time to an MOB interrupt has loose tolerances as explained previously. Filterbank interrupts take precedence to MOB interrupts because the former is the driving 'consumer' process of the logical process pipeline.

An alternative solution is to eliminate filterbank partitions by arranging that the FBP copies all SM buffers with a burst read into its own local memory at the end of the frame whence filterbank execution can be scheduled arbitrarily during the next frame. The initial frame interval, devoted exclusively to MOB-SM traffic, is thus maximised. A problem with this approach is that subband streams pass via the FBP three times rather than once, as occurs in the scheme of Fig. 9.4. A hardware DMA module provides a potential solution. However, transparent DMA transfer using redundant FBP memory cycles would lengthen the SM read burst for which mutual exclusion with the MOB must apply, thus reducing the initial frame interval. An additional problem is that extra DMA hardware increases costs. Therefore the algorithm of Fig. 9.4 was developed which permits random access by the FBP to SM buffers for the filterbank pertaining to each partition under mutual exclusion with the MOB. No extra hardware is required, FBP throughput is not degraded by unnecessary traffic and MOB-SM access is maximised as in the case of a burst read of SM buffers by the FBP, provided that the specified constraints on major run length are obeyed.

9.4 Quantitative Aspects of MASC Implementation

9.4.1 On the Interdependency of MAS Latency and MOB Burst Length

A prototype MASC can only be considered after the attribution of judicious values to the design parameters of K, N_{burst} and T_{frame}. It transpires that eqn. (9.1) is a fundamental relationship which governs how the inherent latency of MAS is traded off with a lower MASC unit cost. The first observation to make is that MOB clock rate may be higher than that for MASC-SM traffic because the former is encapsulated in VLSI and the latter is restricted by SM speed and the electrical properties of the external data-bus (e.g. capacitance). Therefore, N_{burst} is not directly dependent upon the number of SM cycles required for an OD update (six 32-bit word pairs as specified in section 8.5.1); a fact that has a major role in MASC efficiency because a high internal clock rate results in a costlier implementation because a faster VLSI technology is necessary.

From a consideration of the proposed MASC synchronisation algorithm of section 9.3.2, the maximum delay between a MOB interrupt to the Host upon completion of the major run for filterbank j and processing of the accumulation bursts from j by the FBP is $2T_{frame}$ as constituted by the eventuality of a short major run for j that is busy-waiting on filterbank partition j. Filterbank latency $T_{fb}(K)$ from eqn (4.7) must be added and also the delay in Host processing between the arrival of an event to the SME and its translation into MAS parameters, which is a maximum of about T_{frame} when OD schedule update by the Host is on a frame-level MOB interrupt basis (see section 9.3.2) giving a total (worst-case) system latency T_{tot} expressed in eqn (9.2) which emphasises the frame-level synchronicity of the logical process pipeline of Fig. 8.1. Effects processing is not considered because the signal processing applications involved are executed fullband and there are none of the relatively high latencies associated with frame-scheduled multirate operation.

$$T_{tot} = 3T_{frame} + T_{fb}(K) \tag{9.2}$$

On this basis, the optimum value of N_{burst} is determined by combining eqns. (9.2) and (9.3), which includes the latency tolerance T_{max} from section 1.2.2, to give eqn. (9.4). For example, if T_{max}=20ms, K=3 and f_s=44.1kHz and PM-FIR QMF filterbanks with Δ_f=0.1 as justified in section 6.4.4 are used, which have a stage latency of M=23 samples giving $T_{fb}(K)$=3.65ms @ f_s=44.1kHz via eqn (4.7), the result is N_{burst}=30. It is desirable for simple VLSI implementation that the clock frequency ratio between the MASC-SM data-bus and MOB is an integer power-of-two: as traffic on the former is organised as 6 word-pairs per OD, setting N_{burst}=4×6=24 is a sensible choice. T_{frame} must be recalculated by applying eqn. (9.1) with the practical value of N_{burst}. Despite a consequent reduction throughput from quantising N_{burst} thus, lower latency is a positive side-effect: with N_{burst}=24, T_{frame}=4.35ms resulting in T_{tot}=16.7ms via eqn.(9.2). Note that in this case, $T_{fb}(K)$<<T_{tot} showing that latency in MAS is attributable more to the exigencies of MASC optimisation than filterbank delay.

$$T_{max} \geq T_{tot} \Rightarrow T_{max} \geq 3T_{frame} + T_{fb}(K) \Rightarrow T_{frame} = \frac{T_{max} - T_{fb}(K)}{3} \tag{9.3}$$

$$N_{burst} = \mathrm{int}\left(\frac{f_s(T_{max} - T_{fb}(K))}{3(2^K)}\right) \tag{9.4}$$

9.4.2 Quantifying Expected MASC throughput

An approximation of the maximum number of OD’s that can be scheduled per frame by the MASC (if all are allocated in subbands at level K) is given by eqn (9.5). An interesting feature is that there is no K term indicating that max(S) is independent of multirate operation because the quantity is bounded by the MASC-SM bottleneck rather than MOB throughput. The effect of K is manifest in eqn. (9.4) in that N_{burst} is reduced by a substantial amount (there is an approximate proportionality $N_{burst} \propto 2^{-K}$ if $T_{fb}(K) << T_{max}$) which is desirable for two reasons because (i) MOB clock rate is directly proportional to N_{burst} thus permitting the use of cheaper VLSI technologies for MASC implementation and (ii) the range of sample rates provided by MAS obeys Heisenberg's inequality (through extended bursts as discussed in section 9.2.1) which is not provided in a classical AS version of the MASC (i.e. K=0) which only has frame-level breakpoint quantisation c.f. FFT^{-1} (Rodet and Depalle, 1992). For K=3, the MOB operates at 1/8th the rate required at K=0, with the limitation that max(S) is obtainable only in subbands at level K by exploiting the standard *a priori* MAS parameters of $\{f_{min}(x), f_{max}(x)\}$.

$$\max(S) \cong T_{frame} \frac{\text{(MASC - SM data - bus clock freq)}}{\text{(no. of SM accesses per OD)}} \qquad (9.5)$$

It is stated in section 6.4.4 that there is strong evidence for an optimal value of K=3. The insights into MASC implementation in this section also support this point of view. For comparison, setting K= 4 and using the PM-FIR QMF stage prototype for K=3 of section 9.4.1 results in $T_{fb}(K)$=7.82ms making N_{burst}=11 by eqn. (9.4). A practical value of N_{burst}=12 is better for the reasons given earlier. Filterbank cost is increased causing extra processing by the FBP. Quite apart from a severe degradation of subband hierachy caused by the logical exclusion of deadbands, a higher $T_{fb}(K)$ reduces T_{frame} which, in turn, reduces max(S) via eqn (9.5). The benefit of the modification is that MOB clock rate is halved such that it is commensurate with the MASC-SM data-bus. However, this appears to be a false economy because higher clock rates are permissible in the MASC: this latter facility is not exploited and the system becomes bound by MASC-SM bandwidth. At K=3, MOB clock rate is twice the MASC-SM data-bus clock rate assuming a non-interleaved 32-bit SM. It appears reasonable to conclude, though it is desirable to reduce MOB clock rate, that beyond a certain limit economies from using MAS diminish. K=3 is evidently a good operating point.

Consider a hypothetical example of a complete MASC-SM system. Using memory devices with total access times of <100ns (e.g. high speed DRAM's) in an interleaved 32-bit SM (requiring 12 cycles for an OD update) with a data-bus cycle time of 50ns

(20Mhz) and T_{frame}=4.35ms (using the example in section 9.4.1) the result of applying eqn. (9.5) is max(S)≅7256. In this instance, MOB clock rate is 40MHz with N_{burst}=24. Such a figure constitutes AS resources in excess of those required by typical applications (e.g. those discussed in Chapter 6) and raises doubts about the utility of a non-interleaved 64-bit SM scheme as discussed in section 8.5.1. Optimal SM size is the product of max(S) and the size of the prototype OD (aligned at 32 words as discussed in section 8.5.1), plus space for MOB-FBP buffers (see section 9.4.3). In this example, an SM of 907kBytes, rounded to 1MByte, is indicated. However, max(S) is an upper bound and, in typical applications, allocation to subbands at level k<K reduces max(S) which is compounded by FBP and Host SM bandwidth requirements (see section 9.4.3) indicating that there will be a large proportion of SM free-space.

9.4.3 Quantifying the SM Bandwidth Requirement of Host / FBP IPC

Host-SM traffic, which ultimately governs MASC operation, is difficult to quantify. However, some observations can be made. The PWL breakpoint set for an AS tone is usually highly compact (see section 1.1.3). At note onset, the major operation is the initialisation of OD's with sufficient breakpoints in the EC's for the immediate future (i.e. next few frames). So as not to degrade MASC performance more than strictly necessary, Host-SM accesses should be minimised. Also, Host-SM access is more complex than to main memory and may be slower, because of the necessity to suspend MASC-SM traffic beforehand. Host-SM traffic is characterised by bursts of control data upon OD initialisation superimposed upon a steady bandwidth of breakpoints to scheduled OD's. Because of the linked-list schedule, the cost of OD deallocation is minimal; a block of OD's is removed and appended to a free-space list by pointer manipulation of the starting and terminating OD's of the block, regardless of its size.

OD initialisation occurs on the precondition that the MASC can schedule the new OD's in which state there will be redundancy in MASC-SM bandwidth spent 'busy-waiting'. Host initialisation of an OD set can exploit this spare bandwidth without corrupting the processing of scheduled OD's by the MASC because of sub-frame-level asynchronicity between the MOB and FBP as indicated in Fig. 9.4. This is another desirable property of frame-level scheduling, apart from the ease of implementing multirate operations by extended bursts (see section 9.2.1). In the eventuality of MOB-FBP frame-level synchronicity being corrupted by prolonged Host-SM access, the OD schedule will resynchronise within T_{frame} so long the constraints of section 9.3.2 on major run length are satisfied: FBP / MOB mutual-exclusion to SM buffers may be corrupted for a frame causing a 'glitch' in filterbank outputs. However, an efficiently designed Host-SM interface and MASC OS kernel for the SME can minimise this risk by a respective minimisation of MOB suspension and Host-SM access by using resource allocation algorithms dependent on Host-based images of the SM data structure organisation.

Quantifying the bandwidth of MOB-FBP traffic is also imprecise as filterbank number n_{fb} and topology are reconfigurable making the number of subband streams n_{sb} vary widely between different MAS applications. However, the simulations of Chapter 6 enable some speculative results to be derived. In section 6.4.2, n_{fb}=8 with identical instantiations of the prototype of Fig. 6.2: the input bandwidth to each filterbank b_{fb} is simply the sum of all constituent subband sample rates which is given by b_{fb}=f_s+ f_s2^{-1}+2 f_s2^{-2}+4 f_s2^{-3}=2.5 f_s. Therefore at f_s=44.1kHz, b_{fb}=110.25kHz. As the MOB writes accumulation bursts to SM buffers which are subsequently read by the FBP, the total SM bandwidth occupied by MOB-FBP traffic b_{tot} is given by eqn. (9.6) which results in a requirement of b_{tot}=1.764MHz when samples (see section 9.2.2) and the MASC-SM data-bus are 32-bits wide. Considering that the MASC-SM data-bus is anticipated as operating in excess of 20MHz, the proportion of SM bandwidth sacrificed to MOB-FBP traffic is modest and below 10%. Therefore routing MOB-FBP traffic via the SM appears to be highly economic: a principle advantage is that no extra MASC pinout is necessary to support a separate FBP interface. It is relevant also to mention the total space SM occupied by MOB-FBP SM buffers as expressed by eqn. (9.7), which results in n_{buff} =3840 words (using the example of section 9.4.1 where T_{frame}=4.35ms): this is but 1.5% of the 1MByte SM proposed in section 9.4.2.

$$b_{tot} = 2n_{fb}b_{fb} \qquad n_{buff} = T_{frame}n_{fb}b_{fb} \qquad (9.6, 9.7)$$

9.5 Dataflow Aspects of MASC Design

In the light of the preceding analyses of MASC functionality, a systems-level MASC design can now be developed in terms of dataflows between functional units with the objective of identifying the primary internal architectural forms.

9.5.1 Systems-Level MASC Architecture

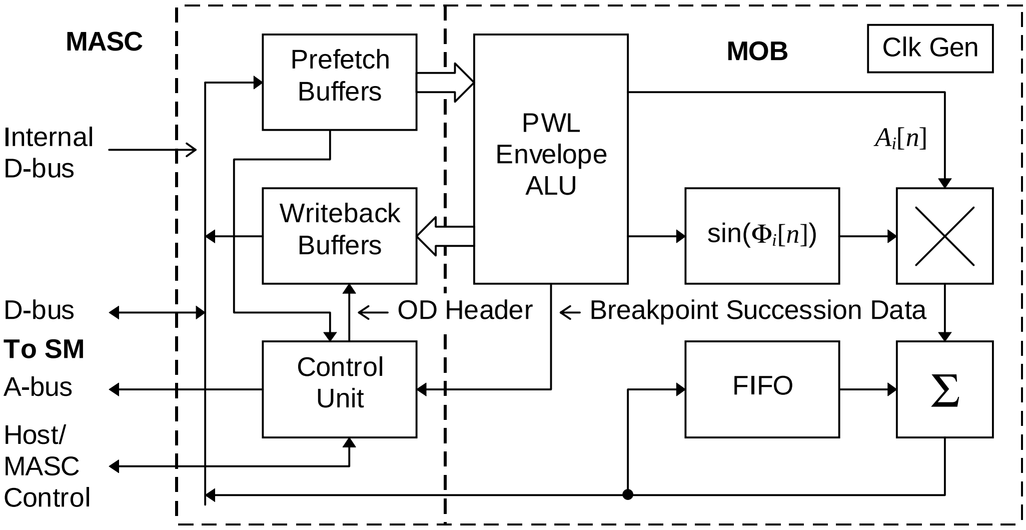

Figure 9.5 Schematic Diagram of MASC Architecture

Fig. 9.5 illustrates how the MOB dominates the architecture of the proposed MASC (Phillips et al, 1996b). When compared to the TOB of section 1.3.2, essential similarities between the dataflow topology can be identified: the distributed state memories of the TOB are lumped together in the MOB into a PWL Envelope ALU described in section 9.5.3. Three major differences are that (i) the MOB operates in a burst mode as specified in section 8.5.2, (ii) multirate operation is implemented by extended bursts in a frame timing structure as proposed in section 9.2.1 and (iii) computation of sin($\Phi_i[n]$) is envisaged as being via a linear interpolated LUT transform which is pipelined for high throughput as discussed in section 7.4.3. Non-MOB MASC functionality is centred around an internal data-bus whose activity is assumed to be identical to that of the external MASC-SM data-bus.

9.5.2 Prefetch and Writeback Buffers

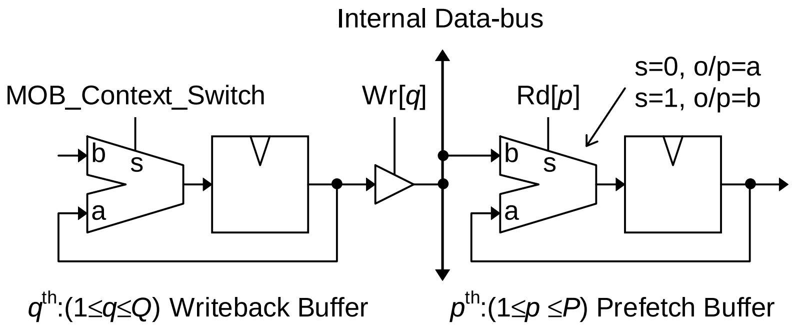

Figure 9.6 Prefetch and Writeback Buffers

Fig. 9.6 illustrates the logic for the Prefetch and Writeback Buffers. The header word also passes through the Control Unit via these buffers. During an OD read, P=8 words are read sequentially into the Prefetch Buffers, by sequencing Rd[1] through to Rd[P] as each OD datum appears on the internal data-bus. The OD is thus demultiplexed for parallel presentation to the PWL Envelope ALU and Control Unit. During breakpoint prefetch in extended MOB bursts, only those buffers pertaining to breakpoints are activated. Before an OD write, MOB_Context_Switch registers the final state of the current OD. Then Q=4 words (i.e. the header word and new accumulator values) are placed on the internal data-bus from the Writeback Buffers by sequencing Wr[1] through to Wr[Q] thus re-multiplexing OD data back to the SM. As discussed in section 8.5.2, these buffers constitute the third level of the MAS-specific memory hierarchy.

9.5.3 PWL Envelope ALU

The PWL Envelope ALU fulfills the two distinct function of (i) uncompression of $A_i[n]$ and $F_i[n]$ and (ii) integration of the phase accumulator ($\Phi_i[n]$) for sine transformation. For the former, two instances of the uncompression logic illustrated in Fig. 9.7 are required. When MOB_Context_Switch is high, a new OD is switched in upon the next MOB clock cycle (these input registers constitute the final fourth level of the MAS-specific memory hierarchy). Otherwise, when low, the accumulator is integrated by the gradient value and compared with the target value so that when undershoot or overshoot (depending on the gradient sign) is detected, the target value itself is substituted for the invalid accumulator value as specified in section 8.5.1. To signify breakpoint succession, New_Breakpoint_Ptr from the target and a Breakpoint_Overflow flag are supplied to the

Control Unit which handles the event. For breakpoint succession in extended bursts, a signal New_Breakpoint retains the current accumulator value, but registers the gradient and target of a new breakpoint from the Prefetch Buffers.

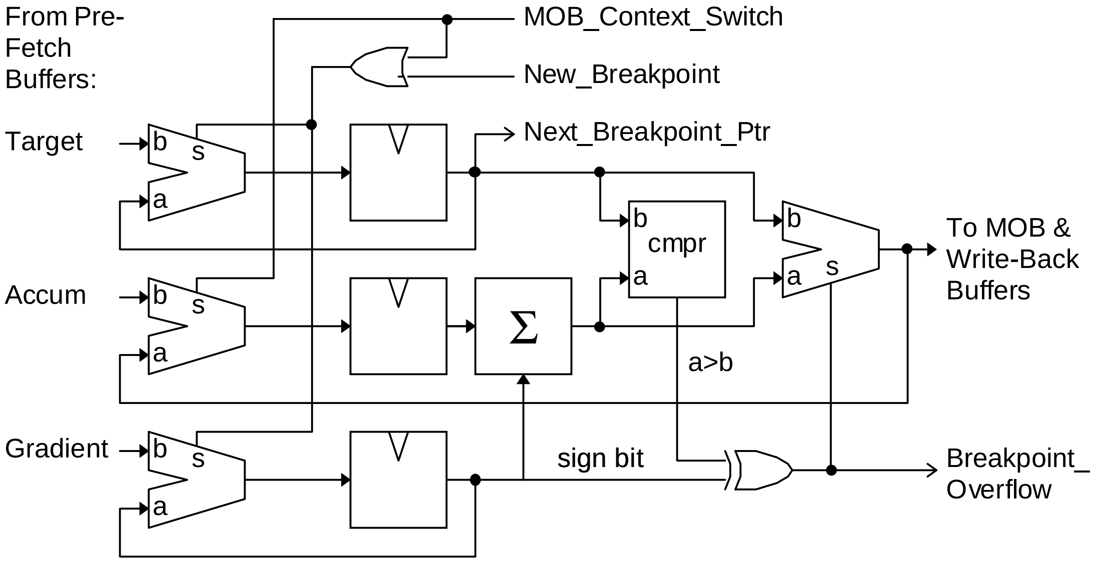

Figure 9.7 Dataflow for $A_i[n]$ and $F_i[n]$ Structures

Discrete integration of the phase accumulator ($\Phi_i[n]$) is rendered more complex by the necessity for trapezoidal integration according to the scheme of section 4.3.2 (bypassed for the case of a fullband OD allocated to k=0). Fig. 9.8 shows the resulting logic which has the same basic form as Fig. 9.7 minus breakpoint handling. $F_i[n\text{-}1]$ and $F_i[n]$ can be conveniently taken from, respectively, the accumulator register and multiplexer outputs in the $F_i[n]$ structure. 2^{-k} multiplication is effected by a simple three-line barrel shifter for K=3. Therefore, ignoring multiplexers, the critical path in the PWL Envelope ALU is (i) an addition and comparison (implemented by a subtraction) in the $F_i[n]$ structure and (ii) three additions in the $\Phi_i[n]$ structure: propagation must be complete in (i) for (ii) to begin. If carry look-ahead is used, and noting the properties of cascaded adders (c.f. array multipliers), throughput bandwidth should be commensurate with succeeding sections of the MOB pipeline. Internal pipelining in the PWL Envelope ALU is therefore not considered necessary as it carries the overhead of startup and flushing which precludes the requirement for instantaneous context switches between OD’s.

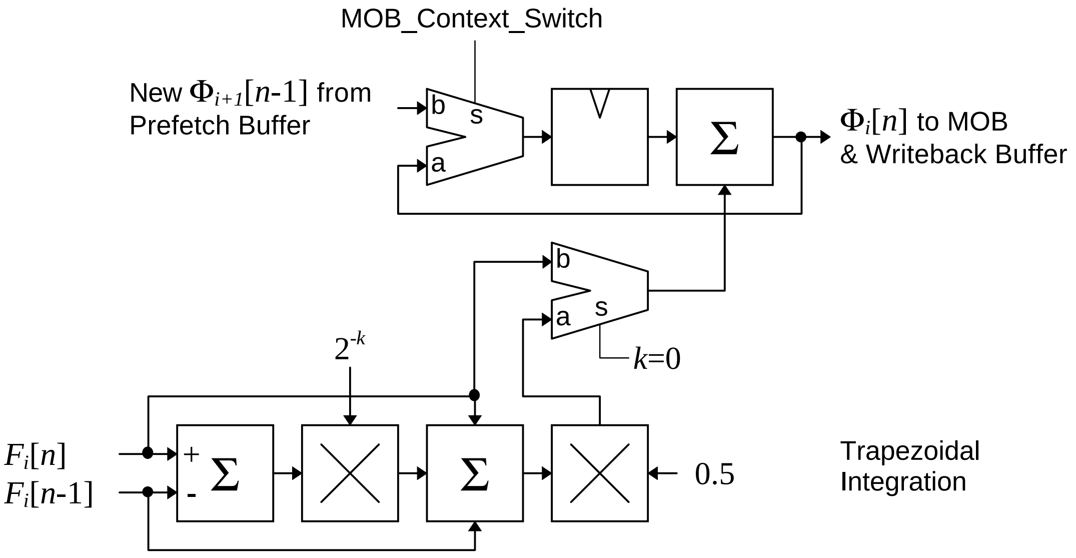

Figure 9.8 Dataflow for $\Phi_i[n]$ Structure

9.5.4 Accumulation Structure

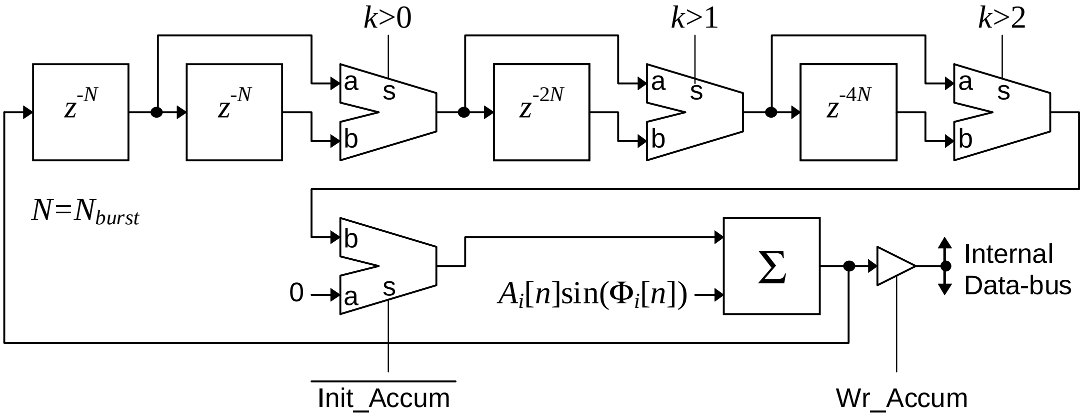

Figure 9.9 Accumulation Structure Dataflow

The accumulation loop FIFO is variable length following the scheme described in section 9.2.2 and is illustrated in Fig. 9.9. Each of the 'k>n' signals represents a truth condition about the level in the subband hierarchy of the current minor run. An input multiplexer to the adder permits a zero addend when Init_Accum is low (implemented by AND gates) so that the FIFO may be initialised during the first MOB burst of a new minor run. When Wr_Accum is high, the accumulated output of the last MOB burst of a minor run is placed in the internal data-bus and written to the MOB-FBP buffers in the SM using an address burst provided by the Control Unit.

9.5.5 Header Word Processing in the Control Unit

Field	#bits	Comment
Next OD pointer	14	(16384 maximum)
Current $A_i[n]$ breakpoint pointer	4	(for the 32-word OD of section 8.5.1)
Current $F_i[n]$ breakpoint pointer	4	ditto
Filterbank Address	4	(16 maximum; c.f. section 6.4.2)
Subband Address k	2	(k=0..3 for K=3)
Subband Address l	3	(l=0..7 at k=K=3)
sync (from section 9.3.2)	1	for frame-level synchronisation

Table 9-1 OD Header Word Bitfield Assignment

The four tasks of the Control Unit are (i) interpretation and updating of header words, (ii) SM address generation, (iii) sequencing of internal MASC dataflows and (iv) SM access arbitration with the Host and FBP (combined in the Type-I topology). All but task (i) require an implementation-level of design but the major facets of functional behaviour are encompassed in this and the preceding chapter. The main operation in header word update is that of detecting a breakpoint succession from the PWL Envelope ALU and replacing the EC pointer for the relevant envelope with the next breakpoint pointer (derived from the target field of the current breakpoint) as outlined in section 8.5.1. The bitfields of the hypothetical 32-bit header word are tabulated in the Table 9-1. The ‘Next OD Pointer’ may be usefully aligned on multiples of 16 words as this permits a minimal OD with an EC of just 6 breakpoints, or longer by multiples of 8 breakpoints as desired. Filterbank and subband addresses (7 bits in total) may be used to address a MOB-FBP buffer pointer table of 128 words (see in section 9.2.2).

9.6 Review

The set of logic designs of MASC functional units provides an appropriate conclusion to the level of abstraction to which the MASC concept is developed in this thesis. A design for the Control Unit in Fig. 9.5 is the major omission. This is because it is dependent upon (i) the low-level organisation of the MASC-SM data-bus and (ii) final decisions about how scheduling and synchronisation may be implemented which can only be made within the context of MASC prototype development. However, the Control Unit will take the form of a state machine which can be synthesised from a lower-level behavioural (as opposed to functional) description of the MASC during simulation. For similar reasons, the mechanism for MASC suspension during Host-SM access is omitted; an efficient solution is to derive the dataflow clock from a higher frequency master clock (e.g. ×2) in a notional Clock Generator circuit for hazard-free gating of the dataflow clock.