The pulse-width modulator is also an easy-to-design block [1]. In its simplest version, it consists of a Schmitt-trigger and a passive loop filter. However, a potential problem with a PWM signal is that it contains many harmonics of fc, the carrier frequency of the PWM. If these harmonics fall near integer multiples of fs, where fs is the clock frequency of the DSM, after sampling in the DSM loop, they might alias to the baseband and corrupt the desired signal [1]. Fortunately, this problem is alleviated in this architecture, because anti-aliasing filtering is one of the inherent qualities of a CT-DSM, and as the order of the modulator increases, the anti-aliasing filtering will improve. As a result, the carrier frequency of the PWM can be chosen relatively freely in the sense that fs can be an integer multiple of fc while the aliasing effects are still modest. Obviously, the concept presented in Fig. 1 can easily be extended to multi-bit conversion by using a multi-bit quantiser and multiple switched current sources in the DAC. It is also clear that the proposed method can be applied to a CT-DSM of arbitrary order.

Design example: To illustrate the effectiveness of the proposed concept, we have applied it to the design of a second-order CT-DSM for a bandwidth of 16 MHz with an OSR of 64 (corresponding to a clock frequency of around 2 GHz) in a standard 65nm CMOS process. Fig. 2 shows the corresponding circuit diagram. The upper part of the Figure is the pulse-width modulator. Conceptually, it consists of a feedback loop with a first-order passive loop filter and a Schmitt-trigger. In the actual implementation, a tapered inverter buffer is added to make the rising and falling edges of the PWM signal steeper. As mentioned in the previous Section, the carrier frequency of the PWM, fc, does not need to be accurately controlled. Nevertheless, there are some considerations on the choice of fc. If fc is too close to the baseband, modulation sidebands may fall partially into the baseband. A high value of fc, on the other hand, increases the power consumption. Considering the targeted bandwidth, a carrier frequency of 500 MHz is chosen for the PWM.

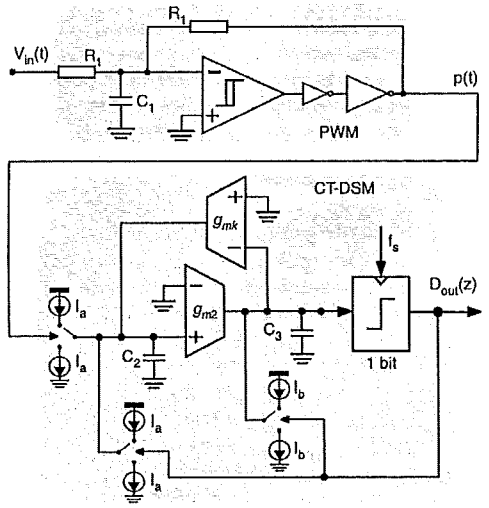

Fig. 2 *Top-level schematic of design example ADC, consisting of pulse-width modulator and g_m-C CT-DSM*

The lower rectangle in Fig. 2 shows the CT-DSM that follows the PWM. The first integrator is implemented by a binary g_m and is driven by the pseudo-digital waveform $p(t)$, as explained above. The requirements on the design of the second integrator, with transconductor g_{m2}, are heavily relaxed because the non-ideal effects of this block are reduced by the high gain of the first integrator and are therefore negligible. For this block a simple source-degenerated differential-pair transconductor was chosen. The local feedback, implemented with the transconductor g_{mk}, has a very small transconductance and has little impact on the overall linearity, noise or power of the modulator. The overall power consumption of the designed circuit is as low as 2.5 mW from a single 1.2V supply.

The resulting output spectrum of a transistor level simulation of this circuit is shown in Fig. 3. The corresponding SNR and SNDR are both equal to 66 dB for a 16 MHz bandwidth. As expected, the PWM carrier and its sidebands are visible around 500 MHz. These tones are safely away from the baseband and they can easily be filtered in the digital domain. The theoretical noise spectrum (calculated from the designed NTF) is also shown in the Figure. It matches the simulation very well, apart from the fact that the notch in the noise spectrum is 'filled' in the simulated spectrum. This is owing to the finite output impedance of the current sources in the first integrator, which can not easily be made arbitrarily large in this 65 nm CMOS technology (because of leakage effects). Owing to this, the DC-gain of the first integrator is not infinite. If this effect is taken into account in the theoretical plot, it matches the experiment nearly perfectly.

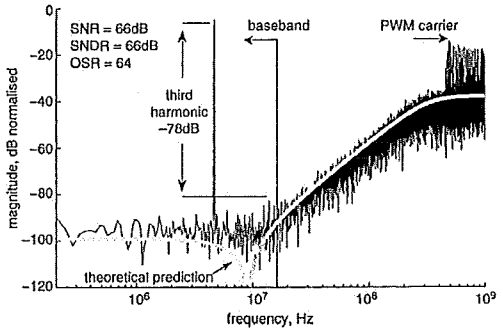

Fig. 3 *Transistor-level simulation result of output spectrum of design example*

Conclusion: This Letter presents a simple technique to implement the first integrator of a CT-DSM with low complexity and power consumption. The approach uses a PWM to convert the input signal into a two-level signal so that a switched current source together with a capacitor can be used as the first integrator in a CT-DSM. The idea can be applied to a CT-DSM arbitrary order that can have a single-bit as well as multi-bit quantiser.

Acknowledgment: This work has been supported by the Fund for Scientific Research Flanders (FWO-Vlaanderen) Belgium.

2 August 2012
doi: 10.1049/el.2012.2697

A.Babaie Fishani and P. Rombouts (*CAS, ELIS, University of Gent, Gent, Belgium)*)

E-mail: amir.babaiefishani@elis.ugent.be

References

1 Hernandez, L., Paton, S., and Prefasi, E.: 'VCO-based sigma delta modulator with PWM precoding', *Electron Lett.*, 2011, 49, (10), pp. 558–559
2 Daniels, J., Dehaene, W., Steyaert, M.S.J., and Wiesbuer, A.: 'A/D conversion using asynchronous delta-sigma modulation and time-to-digital conversion', *IEEE Trans. Circuits Syst.-I*, 2010, 57, (9), pp. 2404–2410
3 De Vuyst, B., and Rombouts, P.: 'A 5-MHz 11-bit self-oscillating sigma delta modulator with a delay-based phase shifter in 0.025 mm^2', *IEEE J. Solid-State Circuits*, 2011, 46, (8), pp. 1919–1927

Convergence time analysis of open shortest path first routing protocol in internet scale networks

S.U. Malik, S.K. Srinivasan and S.U. Khan

Proposed is a novel method to compute the intra-area convergence time of open shortest path first (OSPF) based networks in the presence of designated routers (DRs) on Ethernet and non-broadcast multi-access segments. The capacity of the proposed method is demonstrated by evaluating the convergence time performance of OSFP on internet scale networks (having a thousand autonomous system level routers). The method has also been used to analyse the effects of: (a) the number of DRs, (b) cascading failures, and (c) topological changes on the convergence time of the routers within an area. Furthermore, the time the network takes from a cold state to reach a stable (steady) state in an area is also analysed.

Introduction: Open shortest path first (OSPF) is an adaptive routing protocol to distribute routing information within a single autonomous system (AS) [1]. OSPF divides the network into areas. Each area consists of one or more segments. A segment constitutes the set of routers connected via a common communication channel (example Ethernet). When a failure occurs, topologies are regenerated and paths are recalculated by all of the routers within that area [2]. The time a router takes to discover the area topology is known as the convergence time [1]. To improve the convergence time of a segment in an area, a router is selected as a designated router (DR) on each segment.

Fast convergence time is required to meet network based application demands and quality of service (QoS) requirements of modern dynamic large-scale routing domains, such as data centres. Therefore, a lot of effort and studies have been made to improve the performance of OSPF [3]. However, the convergence time analysis of OSPF that incorporates DRs has never been studied. We address the aforementioned, by developing a novel method to compute the intra-area convergence time of OSPF-based networks that incorporates DRs, which is the primary contribution of this Letter. Moreover, to analyse and benchmark the protocol on internet scale networks is another contribution of this work. We also show how to use our method to study the effect of: (a) DRs, (b) cascading failures, and (c) topological changes on the convergence time of the routers within an area.

For our experiments, we simulated the detailed implementation of the OSPF protocol based on the specifications reported in [2]. To get realistic measurements we generate topologies from BRITE [4], using Otter [4] (as shown in Fig. 1.) that represents the exact same characteristics as those of the internet. The results and analysis provided in this Letter will be extremely useful for network administrators seeking to deploy OSPF. Moreover, the results are also useful in the behavioural analysis of OSPF and can provide the basis to re-evaluate the design of the protocol to achieve performance optimisation.

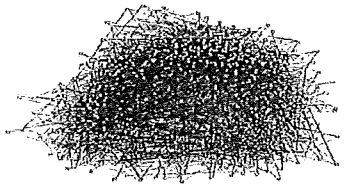

Fig. 1 *Sample topology for one thousand routers*

Problem formulation: Consider a network composed of N routers. Let R_i be the ith router, where $1 \leq i \leq N$. A link between two routers R_i and R_j (if it exists) has a communication cost that represents the minimum time for transferring a message from R_i to R_j, which can be represented by the following expression [5]:

$$del(R_i, R_j) = \frac{D(R_i, R_j)}{\nu} + \frac{s}{\beta_{ij}} \tag{1}$$

where $D(R_i, R_j)$ is the physical distance between R_i and R_j, ν is the propagation delay of the medium (optical fibre in our case), s is the size of the message in kilobytes, and β_{ij} is the available bandwidth between R_i and R_j. If the routers are not directly connected, then the communication cost is the sum of the cost of all links in the shortest path from R_i to R_j. Without loss of generality, we assume that $del(R_i, R_j) = del(R_j, R_i)$, which is a common assumption in the literature [5]. Let M be the number of segments within an area and S^k be the kth segment in that area, where $1 \leq k \leq M$. Let DS be the set of DRs within an area and ς^k is the convergence time of S^k. If a failure occurs (could be a link or a router), the routers connected to the failed link or failed router will initiate the updates. Let R_o^k be a router that initiates an update in response to a failure. Let Rr be the set of all other routers in the area defined as $Rr = (\bigcup_{i=1}^{N}\{R_i\})\backslash\{\{R_o^k\} \cup DS\}$.

R_o^k will detect a failure if no response is received from a neighbouring router for a period longer than the dead interval (DI). R_o^k will then update its link state and forward the updated link state to the DR of segment k (represented as d^k. The link state is the description of the interface of the router (IP address of the interface, mask, type of network, routers connected to) and the relationship to other routers. The DR will then flood the information to every other router in the segment after receiving the update. Let R_i^k represent a router that belongs to S^k. The time for R_i^k to receive the update ($\psi(R_i^k)$) can be calculated as follows:

$$\psi(R_i^k) = \begin{cases} DI, & \text{if } R_i^k = R_o^k \\ \psi_{\forall k \in S: R_i \in k}(d^k) + del(d^k, R_i^k), & \text{if } R_i^k \in Rr \end{cases} \tag{2}$$

where

$$\psi(d^k) = \psi_{\forall j \in k: k \in S}(R_j^k) + del(R_i^k, d^k) \tag{3}$$

We assume that other updates, such as change in bandwidth ($\Delta\beta_{ij}$), are local and incur zero update time. Therefore in (2), the value of $\psi(R_i^k)$ for $R_i^k = R_o^k$, is DI. The DI of routers is usually four times the 'Hello' interval, which is the time between consecutive transmissions of 'Hello' packets that are used to indicate the liveliness of nodes. The 'Hello' interval is 10 seconds for broadcast and P2P networks, and 30 seconds for all other media [2]. The value of $\psi(R_i^k)$ for $R_i^k \in Rr$ is the sum of the time required for d^k to receive updates and the time d^k takes to deliver updates to R_i^k. The value of $\psi(d^k)$ is calculated in (3), which is the sum of $\psi(R_k^k)$ (the node sending the update to d^k) and the communication cost between them, which is given as $del(R_j^k, d^k)$. Moreover, (2) and (3) are used to calculate ς^k based on the following equation:

$$\varsigma^k = max(\psi(d^k) + del_{\forall j \in k}(d^k, R_j^k)) \tag{4}$$

The last router (maximum time taken to receive an update from the corresponding d^k) in S^k that receives the update, determines ς^k. Now, using (2), (3) and (4) the convergence time of an area τ can be calculated as follows:

$$\tau = max_{\forall k \in S}((\varsigma^k)) + \psi((R_o^k) + del(R_o^k, d^k) \tag{5}$$

The maximum ς^k amongst all of the segments plus the time when the update is initiated and reaches to the respective DR determines the value of τ.

Results and discussion: The value of τ determines the time an area requires to reach a stable (steady) state from an unstable state, which is caused by an update. Therefore, to avoid message losses the network must converge quickly. To this end, we evaluate the effect of: (a) the number of DRs, (b) cascading failures, and (c) topology on the value of τ.

We assume optical fibre as the communication medium having propagation delay $\nu = 300 \times 10^6$ miles/s. Ethernet channels have a maximum transmit unit (MTU) of 1500 bytes [1]. Also, fragmentation is usually avoided in OSPF [1]. Therefore, we assume the message size s to be 1 KB (lower than the 1500B cap, but not too low and is typically used in the literature for experimentation, such as in [6]). The bandwidth value β_{ij} is kept constant at 100 Mbit/s, as advocated in [7] for evaluation purposes. The values of $D(R_i, R_j)$ are assigned from within the range of [1–100] km.

Fig. 2 depicts the effect of the number of DRs on the value of τ. To analyse the effect on large and average scale networks, we used $N = \{1000, 300\}$. A DR can decrease the segment convergence time from $O(n^2)$ to $O(n)$ [1]. However, including more DRs in an area has no effect or in some cases may even increase the value of τ. As reported in Fig. 2, the mean value of τ increases gradually as the number of DRs increases in the topology. To see why, consider a router R under DR_1. If DR_2 is added to the area and R now falls under DR_2, then DR_1 can no longer directly communicate with R, but instead it is obligatory to communicate via DR_2. From this example we can see that including a DR can increase the length of communication paths in the area, thereby possibly increasing its convergence time. Therefore, the placement of DR is crucial towards the value of τ.

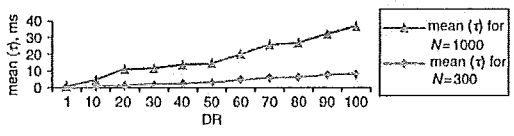

Fig. 2 *Effect of DRs on* τ

Fig. 3 depicts the effect of cascading failures of routers (also called nodes) and links on τ. The number of DRs in an area is set to one (to avoid the influence of multiple DRs on τ). When a node fails, nearby nodes absorb the load of the failed node. The failed nodes can in turn

cause their neighbours to fail (due to overloading) resulting in cascading failures, also known as terminal failure in (communication and power) networks. The degree and placement of a failed node determines its effect on the value of τ. If a failed node or link is in the shortest path of other nodes, then τ may increase. This is because updates to such routers may require a longer path. However, if a failed node or link is: (a) a leaf node, a node with low degree, or link on the edge of the topology, or (b) not included in the shortest path, then τ can decrease as the failed node need not be updated. Moreover, as can be seen from Fig. 3, node failures can affect τ more adversely than link failures. Link failures directly affect only the two routers they are connected to, but node failures affect all its neighbours which is typically more than two.

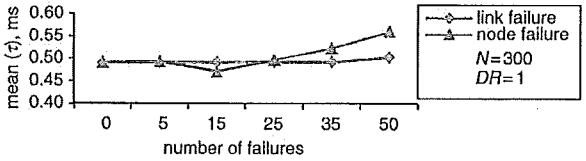

Fig. 3 *Effect of node and link failure on* τ

Fig. 4 illustrates the variation of τ due to the changes in the topology, and shows that the value of τ increases as the number of routers within an area increases. However, an interesting observation is that the value of τ may decrease in certain cases when a router that is included in a topology changes the value of $max(s^k)$ by adding a new shortest path. To avoid the influence of multiple DRs in an area on τ, the number of DRs is set to one in Fig. 4.

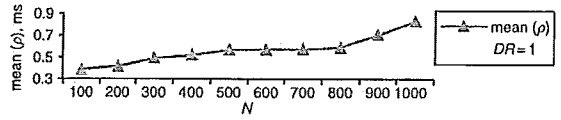

Fig. 4 *Effect of random topological changes on* τ

Conclusion: We have developed a method to compute the intra-area convergence time for OSPF-based networks that incorporates DRs. The results reveal that the factors, such as: (a) the number of DRs, (b) the placement of DRs, (c) interconnection amongst the routers, and (d) the number of routers, in a topology impact the convergence time. Our convergence time analysis technique can therefore be used to evaluate and compare OSPF network design solutions.

5 July 2012
doi: 10.1049/el.2012.2310
One or more of the Figures in this Letter are available in colour online.
S.U. Malik, S.K. Srinivasan and S.U. Khan (*North Dakota State University, ND 58108, USA*)

E-mail: saif.rehmanmalik@ndsu.edu

References

1 Moy, J.T.: 'OSPF; Anatomy of an Internet routing protocol' (Addison-Wesley, 1998)
2 Moy, J.: 'OSPF Version 2, The Internet Society OSPFv2', http://www.ietf.org/rfc/rfc2328.txt, accessed on 08 May, 2012
3 Goyal, M., Soperi, M., Baccelli, E., Choudhury, G., Shaikh, A., and Hosseini, H.: 'Improving convergence speed and scalability in OSPF: a survey', *IEEE ComST*, 2012, **14**, (2), pp. 443–463
4 Medina, A., Lakhina, A., Matta, I., and Byers, J.: 'BRITE: An Approach to Universal Topology Generation', MASCOTS, Ohio, 2001
5 Khan, S.U., and Ahmad, I.: 'A pure Nash equilibrium based game theoretical method for data replication across multiple servers', *IEEE Trans. Knowl. Data Eng.*, 2009, **21**, (4), pp. 537–553
6 Wang, B., Zhang, J., Guo, Y., and Chen, W.: 'Fast-converging distance vector routing mechanism for IP networks', *J. Netw.*, 2010, **05**, (9), pp. 1069–1075
7 Prasad, R.S., Murray, M., Dovrolis, C., and Claffy, K.: 'Bandwidth estimation: metrics, measurement techniques, and tools', *IEEE Netw.*, 2003, **17**, (6), pp. 27–35

Design of latch-based C-element

J.P. Murphy

The C-element logic gate is a key component for constructing asynchronous control in silicon integrated circuits. The purpose of this reported work is to introduce a new speed-independent C-element design, which is synthesised by the asynchronous PETRIFY design tool to ensure it is composed of sequential digital latches rather than complex gates. The benefits are that it guarantees correct speed-independent operation, together with easy integration in modern design flows and processes. It is compared to an equivalent speed-independent complex gate C-element design generated by PETRIFY in a 130 nm semiconductor process.

Introduction: Asynchronous controllers in silicon integrated circuits (ICs) sequence operations arbitrate between events and communicate with their environment through data-flow rather than by clock signals. This flexibility, together with their innate resilience to nanoscale process, voltage and temperature (PVT) variations, allows them to readily address power and clock distribution challenges. As such, they are widely employed in modern ICs for constructing elastic pipelines, building local clock generators or for enhancing data security [1].

In a synchronous IC block, such as a fast Fourier transform (FFT) coprocessor, an asynchronous controller would usually implement the equivalent functionality of the clocking infrastructure and the finite state machine (FSM). This would form typically between 5 and 15% of the asynchronous IC block version's total area. Internally asynchronous controllers use one or more C-element gates to implement speed-independent circuitry that is glitch-free regardless of gate and wire delays. C-elements were originally invented by Muller [2] and are often referred to as the 'Muller C-element'.

A textbook C-element has two inputs, A and B, and one output, Z, with logical behaviour as follows: the output will transition to equal the value of the two inputs after both inputs have transitioned to the same value; otherwise, the output will remain unchanged. They differ from standard logic gates (i.e. NAND/NOR) due to the use of hysteresis (feedback), which allows their logical state to be statically held. As a rule, this is implemented by an inverter bistable appended to the output at the transistor-level, or via wiring the output back to form one of the inputs at the gate-level.

The C-element symbol, transistor implementation and two-input gate implementation are shown in Figs 1*a–e*, respectively. Although, the transistor-level implementation is the most economical in terms of silicon area, such C-element designs are not available in semiconductor digital libraries or field programmable gate arrays (FPGAs). In addition, in nanoscale semiconductor processes the inverter bistable may become unstable due to PVT effects and sub-threshold voltage operation [3].

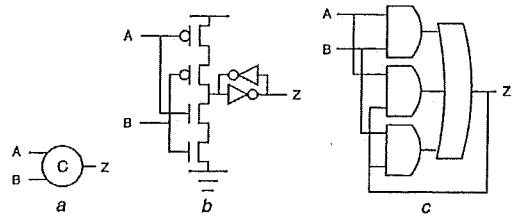

Fig. 1 *C-element symbol and two common designs*
a Symbol
b Transistor-level C-element
c Gate-level C-element

While in the gate-level implementation, the C-element's hysteresis forms a combinational timing loop, which makes rapid design and prototyping of IC blocks difficult in modern design flows and tools. The reason is that such tools are engineered to understand only straightforward latch-logic-latch timing relationships and hardware description language (HDL) code. Therefore, for gate-level C-element designs, coding and modelling in HDLs; synthesis with semiconductor digital libraries and FPGAs; timing analysis; and implementation of testability features is, in general, either error-prone or impossible. Furthermore, the gate-level implementation of Fig. 1*c* is not purely speed-independent, because one of the internal nodes may glitch due to wire delays – a