Enabling Scheduling Analysis of Heterogeneous Systems with Multi-Rate Data Dependencies and Rate Intervals

Marek Jersak, Rolf Ernst
Technical University of Braunschweig
Institute of Computer and Communication Network Engineering (IDA)
D-38106 Braunschweig, Germany
{jersak,ernst}@ida.ing.tu-bs.de

ABSTRACT

Formal methods are growing in importance for performance analysis of real-time systems, but embedded system heterogeneity limits the application of these methods to subsystems or special cases. One of the problems is the rich variety of interactions between embedded system processes, which cannot be directly expressed with the typical event models used in real-time analysis.

This paper shows how to transform complex interaction patterns into the integral representation of minimum and maximum arrival curves, and then to conservatively approximate these arrival curves using standard event models. This approach paves the way to apply the formal approaches known from real-time analysis to heterogeneous embedded systems.

Categories and Subject Descriptors

C.3 [**Computer Systems Organization**]: Special-Purpose and Application-Based Systems—*real-time and embedded systems*; C.4 [**Computer Systems Organization**]: Performance of Systems

General Terms

Theory, Algorithms, Performance, Verification

Keywords

Heterogeneous Embedded Systems, Real-Time Systems, Multi-Rate Data Dependencies, Rate Intervals, Multiple Activating Inputs, Scheduling Analysis, Event Models, Arrival Curves

1. INTRODUCTION

With growing system complexity, formal methods for system performance analysis and estimation receive more attention as an alternative or complement to an increasingly expensive performance

simulation. Formal methods have successfully been used in real-time system analysis (e.g. schedulability analysis), but embedded system heterogeneity limits the application of these methods to subsystems or special cases. One of the problems is the rich variety of interactions between embedded system processes. While real-time analysis usually assumes simple process dependencies, such as acyclic task graphs and single-rate data dependencies, embedded system functions include multi-rate data flow graphs or conditional communication leading to data rate intervals. The resulting interaction patterns cannot be directly expressed with the typical event models used in real-time analysis.

Little previous work addresses this issue. In [6], bursts of events arriving at a process input have a common deadline and lead to the production of one output event. However, this does not solve the problem of data rate transitions between sender and receiver, and the approach is only applicable to earliest-deadline first (EDF) scheduling. Since no output event models are calculated, heterogeneous multi-processor systems cannot be analyzed. An extension presented in [15] allows to analyze single-processor EDF scheduling of synchronous dataflow graphs (SDF) [8]. This is done by transforming the multi-rate SDF graph into an equivalent single-rate representation and calculating individual deadlines for each process activation. Again, since no output event models are used, the approach is restricted to single component analysis.

This paper shows how to transform complex interaction patterns into the integral representation of minimum and maximum arrival curves, and then to conservatively approximate these arrival curves using standard event models. This approach paves the way to apply the formal approaches known from real-time analysis to heterogeneous embedded systems. Our approach is motivated by the work presented in [9], where event models are propagated to combine existing single-component timing analysis techniques, in order to analyze heterogeneous systems for which no coherent analysis technique exists. However, only single-rate systems with single enabling inputs are considered.

The remainder of the paper is organized as follows: In the next section we describe typical embedded system properties that can be captured using our approach. This is followed in Sec. 3 by an overview of standard models employed by existing analysis techniques. In Sec. 4 we explain how to transform complex interaction patterns into existing event models, first for multi-rate data depen-

dencies, then for data rate intervals and finally for multiple activating inputs. The paper concludes with a summary and a brief outlook on future work.

2. SYSTEM PROPERTIES

State of the art embedded systems consist of multiple heterogeneous processing and communication components, either highly integrated in the form of multi-processor systems on chip (MP-SoC) or as distributed systems. They execute a multitude of different parallel functions, many of them with hard real-time constraints. Process scheduling is controlled by real-time operating systems (RTOS) with different scheduling strategies, and communication scheduling happens through a variety of bus arbitration protocols.
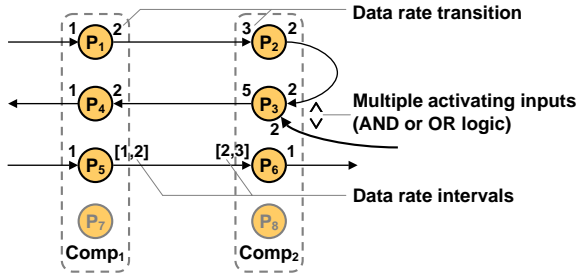

Figure 1: System example

In our system example in Fig. 1, six processes (P_1 through P_6) have been mapped to two different processing components. The processes are scheduled by an RTOS together with an arbitrary number of other processes, represented by P_7 and P_8. We do not restrict the exact scheduling policies or the scheduling parameters. Arbitration of shared communication resources is neglected for simplicity in the example and is not a restriction of our approach.

Realistic systems exhibit multi-rate data dependencies between processes, e.g. in dataflow graphs [8], as well as data rate intervals at process inputs and outputs and multiple activating inputs per process. Our example exhibits a data rate transition between P_1 and P_2 from a smaller production rate to a larger consumption rate, resulting in an execution rate transition from a faster to a slower rate. The opposite happens between P_3 and P_4. The production and the consumption rates between P_5 and P_6 are intervals. P_3 additionally has 2 activating inputs that can be concatenated with either *and* condition, i.e. the process is activated when sufficient data has arrived at all inputs, such as in dataflow graphs, or with *or* condition, i.e. sufficient data has arrived at least at one of the inputs, e.g. in FSM networks. A detailed discussion of *and* and *or* activation and their impact on execution rates can be found in [2].

3. EXISTING SCHEDULING ANALYSIS MODELS

Scheduling analysis techniques typically assume that processes are activated by a stream of arriving events. The minimum and

maximum number of arriving events within a certain time interval is bounded, which can be efficiently expressed with so called *event models*. Using these event models, as well as the core execution time of each process and assigned scheduling parameters, a scheduling analysis for a component can calculate the load of that component as well as minimum and maximum response times for each process scheduled on that component. This allows to validate deadlines, for example. Overviews of analysis techniques for single processors are given in [1, 4].

Event models are generally categorized as periodic or sporadic and additionally can display jitter or bursts [12]. In the following we use the *periodic with jitter* event model as an example to illustrate our approach, but this is not a restriction. The *periodic with jitter* event model states that each event generally arrives periodically with period $\mathcal{P}$, but that it can jitter around its exact position within a jitter interval $\mathcal{J}$. For example, the minimum and maximum distance between two events are

$$d_{min} = \mathcal{P} - \mathcal{J}, \ \ d_{max} = \mathcal{P} + \mathcal{J}$$

$\mathcal{J}$ can be equal or larger than $\mathcal{P}$. If $\mathcal{J} = \mathcal{P}$ then 2 events can arrive at the same time, while the earliest arrival time of the 3rd event is one period later. For a larger jitter, the 3rd event can arrive earlier. If $\mathcal{J} = 2 * \mathcal{P}$ then three events can arrive at the same time while the earliest arrival time of the 4th event is one period later, and so on.

Event sequences can be represented as integrals over time using event model functions [5] or *arrival curves* [11]. Time integrals give nice graphical representations and fit the load analysis approach to scheduling analysis [4], but finding a closed form can be difficult. One approach presented in [11] is to introduce a new analysis technique, where the arrival curves are approximated with piecewise linear functions for the maximum and minimum number of arriving events. In this paper, we propose to extract model properties from arrival curves such that the the upper and lower arrival curves can be represented by the event models used in classical real-time analysis. This way, we are able to make use of existing powerful analysis techniques that have been developed in the field of real-time analysis.

DEFINITION 1. *For any Δt, the upper arrival curve is a tight upper bound for the number of events that can arrive during any interval of length Δt, while the lower arrival curve is a tight lower bound for the number of events that must arrive during any interval of length Δt.*

In Fig. 2, upper and lower arrival curves are shown for the *periodic with jitter* event model with $\mathcal{P} = 4$, $\mathcal{J} = 1$.

Recently, it has been shown how event models can be used to couple analysis techniques for different processors and buses in an MP-SoC or a distributed system, for which no coherent scheduling analysis is available [9, 10]. Single component analysis is performed using existing techniques. The analysis is extended to produce output event models which are propagated to the next component, where they serve as input event models for the analysis of that component. It may be necessary to adapt an event model to suite the analysis requirements for the receiving component, e.g.
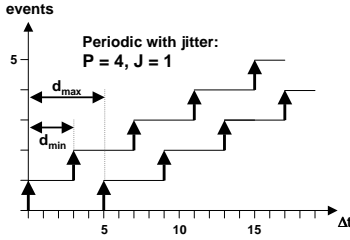
Figure 2: Arrival curves

by reducing the maximum jitter through controlled buffering [10, 14]. All put together, this approach allows to analyze heterogeneous multi-processor systems.

However, existing component analysis techniques usually assume single-rate data dependencies between processes. Consequently, event streams are a sequence of single events, and each event results in exactly one process activation. This is a serious restriction for the analysis of realistic systems, which exhibit multi-rate data dependencies, data rate intervals and multiple activating inputs as shown in Fig. 1. In case of multi-rate data dependencies, tokens produced by one activation of the producing process may not be sufficient to activate the consuming process, or, on the contrary, may result in more than one activation of the consuming process. With data rate intervals, the identification of single activating events is even less clear. Processes with multiple activating inputs are also not supported by most existing timing analysis techniques. Here, a notable exception is recent work on the timing analysis of conditional process graphs [3, 13]. However, these workings assume holistic, homogeneous scheduling of multi-processors and thus do not support the coupling of different analysis techniques in a heterogeneous system with heterogeneous scheduling strategies.

In the following, we show how the aforementioned restrictions of existing timing analysis techniques can be abolished.

4. ANALYSIS MODEL EXTRACTION

In this section we show how to extract standard event models which are directly applicable for coupling existing timing analysis techniques from realistic systems which exhibit data rate transitions, data rate intervals and multiple activating inputs. We illustrate our approach using the example in Fig. 1 and assume that all relevant event models are of type *periodic with jitter* which was introduced in Sec. 3. However, our methodology is equally applicable to other event models.

4.1 Data Rate Transitions

We first consider the data rate transition from a smaller production rate to a larger consumption rate, using processes P_1 and P_2 in Fig. 1 as an example. Let us assume that P_1 produces events with the following properties at its output:

$$\mathcal{P}(P_1) = 4, \quad \mathcal{J}(P_1) = 1$$

The upper and lower arrival curves of this event model are shown in Fig. 3 a). However, different to Fig. 2, we no longer integrate over

the number of events, but instead integrate over the number of communicated tokens. In case of producing processes, we assign each event a height corresponding to the number of tokens produced per activation, in this case 2. Accordingly, we model the activation of the consuming process as events with a height corresponding to the number of tokens consumed per activation. For this to work, we have to re-map events of height r_P into events of height r_C and calculate the resulting event model.

The key idea is that independent of their height, events are treated as atomic from the perspective of scheduling analysis, and thus each consumed event results in one activation. Therefore, standard scheduling analysis techniques that require activation by single events become applicable. Of course, the number and size of tokens produced and consumed is still important for the dimensioning of communication buffers.
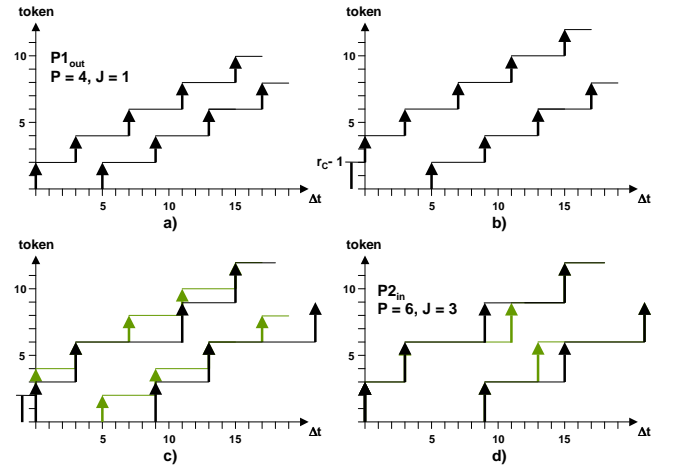

Figure 3: Data rate transition from a smaller production rate to a larger consumption rate.

To correctly construct the arrival curves of the consuming process P_C, we must consider between 0 and $r_C - 1$ initial tokens at the input of P_C [1]. In our example, $r_C(P_2) = 3$. The maximum number of arriving tokens for any Δt at the input of P_C is obtained if $r_C - 1$ initial tokens and the earliest possible arrival times for tokens from the producing process P_P are assumed. Therefore, the upper arrival curve of P_P is shifted upwards by $r_C - 1$ as shown in Fig. 3 b).

The upper and lower arrival curves of the consuming process are constructed from events with a height corresponding to the number of tokens consumed per activation of the consuming process. They bound the minimum and maximum number of activations of the consuming process for any time interval of length Δt. The curves can never be higher than the respective curve of the producing process (or else non-existent tokens would be consumed). This is shown in Fig. 3 c).

At this point, all that remains is to find parameters for some event model to either exactly describe or conservatively approximate the

---

[1]More initial tokens do not have to be considered since they would have already activated P_C in the past.

upper and lower arrival curves of the consuming process. An exact description can be obtained using e.g. the formalism proposed in [5]. However, an exact description can be rather complex, and the complexity is likely to rise as event models get propagated through the system. With a conservative approximation on the other hand, event model complexity can be bounded.

It is possible to obtain an approximation for the consuming process P_C using an event model of the same class that was used for the producing process P_P, in our example *periodic with jitter*. The target event model parameters are calculated as follows:

$$\mathcal{P}(P_C) = \mathcal{P}(P_P) * \frac{r_C}{r_P}$$
$$\mathcal{J}(P_C) = max(\mathcal{J}_{upper}(P_C), \mathcal{J}_{lower}(P_C))$$
$$= max(\mathcal{P}(P_C) - d_{Cmin}, d_{Cmax} - \mathcal{P}(P_C))$$

where d_{Cmin} and d_{Cmax} are the minimum and maximum distances between two activations of P_C

$$d_{Cmin} = max(n_{Pmin} * \mathcal{P}(P_P) - \mathcal{J}(P_P), 0)$$
$$d_{Cmax} = n_{Pmax} * \mathcal{P}(P_P) + \mathcal{J}(P_P)$$

Here, n_{Pmin} and n_{Pmax} are the minimum and maximum number of activations of P_P needed between two activations of P_C.

$$n_{Pmax} = \left\lceil \frac{r_C}{r_P} \right\rceil$$

n_{Pmin} is obtained by first determining the maximum number of tokens δ_{max} possibly available after an activation of P_C

$$\delta_{max} = r_C - 1 + r_P - r_C = r_P - 1$$

and then calculating

$$\delta_{max} + n_{Pmin} * r_P \geq r_C$$
$$\Leftrightarrow n_{Pmin} = \left\lceil \frac{r_C - r_P + 1}{r_P} \right\rceil$$

This is shown in Fig. 3 d) for our example, where

$$\mathcal{P}(P_2) = \mathcal{P}(P_1) * \frac{r_2}{r_1} = 4 * \frac{3}{2} = 6$$
$$\mathcal{J}(P_2) = max(6 - 3, 9 - 6) = 3$$

As can be seen, the approximation is not overly conservative. The shortest Δt for the arrival of $3, 5, \ldots$ events is underestimated by two time units, while the longest Δt between the arrival of $3, 5, \ldots$ events is overestimated by two time units. An extension usually yielding higher precision would be to maintain separate values for $\mathcal{J}_{upper}(P_C)$ and $\mathcal{J}_{lower}(P_C)$. We will return to this in Sec. 4.2.

We now consider the data rate transition from a larger production rate to a smaller consumption rate, using P_3 and P_4 in Fig. 1 as an example. Let us assume an output event model from P_3 with the following properties:

$$\mathcal{P}(P_3) = 5, \quad \mathcal{J}(P_3) = 2$$

Fig. 4 a) shows the upper and lower arrival curves as produced by process P_3, with the upper curve shifted upwards by $r_4 - 1 = 1$ in Fig. 4 b) in analogy to Fig. 3 b). The curves are overlayed in Fig. 4 c) with the upper and lower arrival curves as consumed by process P_4 in analogy to Fig. 3 c).
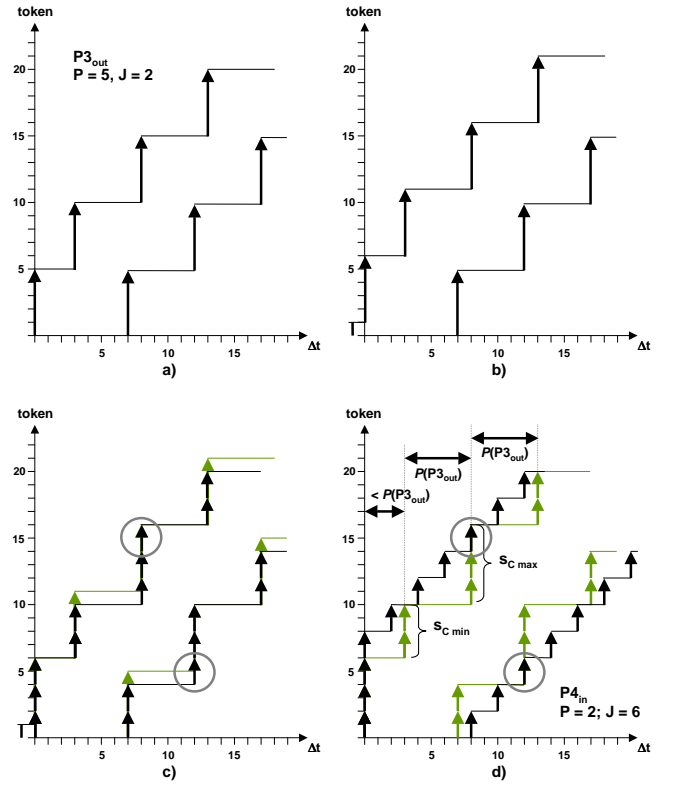

Figure 4: Data rate transition from a larger production rate to a smaller consumption rate.

For all data ratios with $r_P/r_C > 1$, one activation of the producing process can lead to more that one activation of the consuming process. The minimum and maximum number of simultaneous activations of the consuming process is

$$s_{Cmin} = \left\lfloor \frac{r_P}{r_C} \right\rfloor, \quad s_{Cmax} = \left\lceil \frac{r_P}{r_C} \right\rceil$$

This kind of bursty behavior can be conservatively approximated using a *periodic with jitter* event model, where the jitter is equal or larger than the period as explained in Sec. 3.

The period of P_C is calculated as before:

$$\mathcal{P}(P_C) = \mathcal{P}(P_P) * \frac{r_C}{r_P}$$

To calculate the minimum jitter required for a conservative approximation, we need to find a 'critical' activation in the upper and lower arrival curves of P_C. 'Critical' activations are the highest 'peaks' above and the deepest 'valleys' below the average slope of the curves. 'Critical' activations with the smallest Δt_{crit} are circled in Fig. 4 c). The goal is to tightly bound 'critical' activations in the conservative approximation as shown in Fig. 4 d).

We show the calculation of the jitter $\mathcal{J}_{upper}(P_C)$ that results from the upper arrival curve. The calculation of $\mathcal{J}_{lower}(P_C)$ for the lower curve is similar and is given in [7].

As explained before, the upper arrival curve is constructed assuming the maximum number of initial tokens. Therefore, $\Delta t = 0$ is the starting point of the longest possible sequence of s_{Cmax} simultaneous activations of P_C. If this sequence is longer than 1,

then the smallest Δt_{crit} is the value of Δt for the last element in this sequence (each element rises higher above the average slope than the previous one). If the length of the sequence equals 1, then this approach would return $\Delta t = 0$, which due to the jitter may not be critical. However, the smallest $\Delta t > 0$ for which $s_C = s_{Cmax}$, will be critical as shown in Fig. 4 d).

The number of activations in the upper arrival curve of the target *periodic with jitter* event model after the initial burst is

$$n_a = \left\lceil \frac{\Delta t_{crit}}{\mathcal{P}(P_C)} \right\rceil$$

The earliest Δt of the first event after the initial burst in the target event model is

$$\Delta t_{1st} = \Delta t_{crit} - (n_a - 1) * \mathcal{P}(P_C)$$

Let δ_{crit} be the maximum number of tokens consumed for any time interval $\Delta t = \Delta t_{crit}$. The number of activations in the upper arrival curve during the initial burst in the target event model is

$$n_d = \frac{\delta_{crit}}{r_C} - n_a$$

Consequently, the jitter resulting from the upper arrival curve is

$$\mathcal{J}_{upper}(P_C) = (n_d - 1) * \mathcal{P}(P_C) + \mathcal{P}(P_C) - \Delta t_{1st}$$

and, as before, the resulting output event model jitter is

$$\mathcal{J}(P_C) = max(\mathcal{J}_{upper}(P_C), \mathcal{J}_{lower}(P_C))$$

In our example, $s_{Cmin} = \lfloor 5/2 \rfloor = 2$, $s_{Cmax} = \lceil 5/2 \rceil = 3$, $n_{crit} = 2$. Thus

$$\begin{aligned}
\Delta t_{crit} &= 8 \\
n_{Tcrit} &= 16 \\
\mathcal{P}(P_4) &= 5 * 2/5 = 2 \\
n_a &= \left\lceil \frac{8}{2} \right\rceil = 4 \\
\Delta t_{1st} &= 8 - (4 - 1) * 2 = 2 \\
n_d &= \frac{16}{2} - 4 = 4 \\
\mathcal{J}_{upper}(P_4) &= (4 - 1) * 2 + 2 - 2 = 6
\end{aligned}$$

Since $\mathcal{J}_{upper}(P_4) = \mathcal{J}_{lower}(P_4)$ in this example, the resulting *periodic with jitter* event model parameters are

$$\mathcal{P}(P_4) = 2, \quad \mathcal{J}(P_4) = 6$$

As a final note, for the special case of equal output and input data rates, both calculations presented yield the same activating event model for the consuming process as is output by the producing process [7], as would be expected.

4.2 Data Rate Intervals

The transition between data rate intervals builds on the transition between fixed data rates explained in the previous section with the following extensions: to construct the upper arrival curve for consumed tokens, the maximum production rate, the minimum consumption rate and the maximum number of initial tokens at the input of P_C are used; to construct the lower arrival curve for consumed tokens, the minimum production rate, the maximum consumption rate and zero initial tokens at the input of P_C are used. Note that without additional information a lower consumption rate of zero is problematic because a bounded buffer cannot be guaranteed.

Obviously, two event models with different periods are now required to conservatively bound the upper and lower arrival curves. This additional complexity can easily be handled when event models are propagated through the system for analysis. Worst case load is calculated based on the upper arrival curve models, while best case load is calculated based on the lower arrival curve models.

Before we present the construction process, we have to address an interpretation issue regarding the minimum number of tokens required at the input of P_C for P_C to be able to execute. Our interpretation is as follows: P_C requires its maximum consumption rate of tokens available at its input to execute. This is because the total number of tokens consumed may depend on the values of the first tokens consumed, and we do not want P_C to stall for lack of tokens. Our approach remains equally valid for other interpretations of P_C's activation condition.
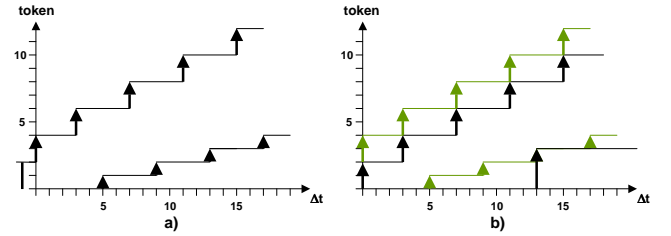

Figure 5: Arrival curves for produced and consumed tokens for processes with data rate intervals.

An example are processes P_5 an P_6 in Fig. 1. The corresponding arrival curves are shown in Fig. 5, with Fig. 5 a) constructed in analogy to Fig. 3 b) and Fig. 5 b) constructed in analogy to Fig. 3 c). Note that due to our interpretations of P_C's activation condition, at least 3 tokens are required at its input for execution. The two bounding *periodic with jitter* event models can be constructed as explained in Sec.4.1.

4.3 Multiple Inputs

If a process has two inputs, than activation typically either requires a minimum number of tokens at at least one input (*or* condition), or a minimum number of tokens at both inputs (*and* condition). More than two activating inputs allow to combine these two possibilities. In case of *and* condition, both data rates have to be the same, otherwise one buffer cannot be bounded.

The upper and lower arrival curves for *or* condition are the respective sums of the two upper and the two lower arrival curves of the two input event models. The upper and lower arrival curves for *and* condition are the lesser of the two upper and the two lower arrival curves of the two input event models, respectively. This is shown in Fig. 6 for two *periodic with jitter* event models with equal periods and different jitter. Note that the input event model with the

smaller jitter contributes the upper arrival curve for *and* condition, while the input event model with the larger jitter contributes the lower arrival curve. For *or* condition, a bounding event model can be constructed as explained in Sec.4.1.
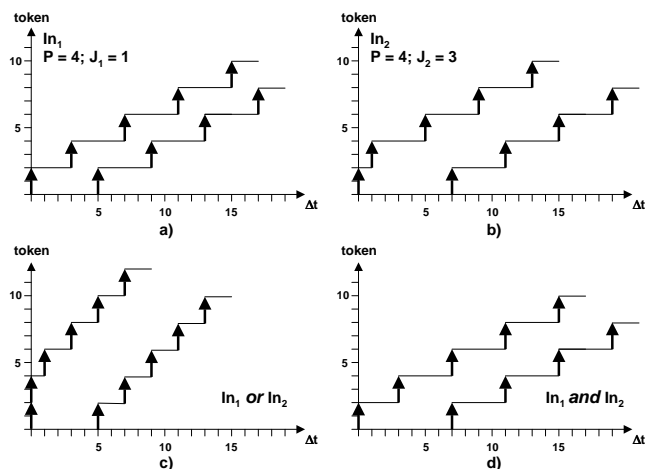

Figure 6: Arrival curves for two activating inputs with c) *or* **and d)** *and* **condition.**

5. CONCLUSIONS

In this paper we showed how to extract standard single-rate event models from realistic systems which exhibit multi-rate data dependencies, data rate intervals and multiple activating inputs. The transformation is performed by first obtaining minimum and maximum arrival curves, and then conservatively approximating them using standard event models. The resulting event models are directly applicable to existing scheduling analysis techniques, thus enabling scheduling analysis of systems that so far were out of reach for those existing techniques.

A major benefit of event models is their ability to couple existing analysis techniques for single components, in order to enable timing analysis of complex, heterogeneous MP-SoC and distributed systems for which no single coherent analysis exists. As a result of the transformations presented in this paper, this coupling has been taken beyond single-rate systems. We are currently investigating the combined benefits resulting from first transforming complex interaction patterns into single-rate event models, and then using those event models to couple analysis techniques.

6. REFERENCES

[1] G. Buttazzo. *Real-Time Computing Systems - Predictable Scheduling Algorithms and Applications*. Kluwer Academic Publishers, 2002.

[2] A. Dasdan, D. Ramanathan, and R. K. Gupta. A timing-driven design and validation methodology for embedded real-time systems. *ACM Trans. on Design Automation of Electronic Systems*, 3(4), Oct. 1998.

[3] P. Eles, K. Kuchcinski, Z. Peng, P. Pop, and A. Doboli. Scheduling of conditional process graphs for the synthesis of embedded systems. In *Proc. Design, Automation and Test in Europe - DATE*, 1998.

[4] C. J. Fidge. Real-time schedulability tests for preemptive multitasking. *Real-Time Systems*, 14:61–93, 1998.

[5] K. Gresser. *Echtzeitnachweis ereignisgesteuerter Realzeitsysteme*. PhD thesis, Technische Universität München, 1993.

[6] K. Jeffay and S. Goddard. A theory of rate-based execution. In *IEEE Real-Time Systems Symposium*, 1999.

[7] M. Jersak. Extending existing scheduling analysis techniques to heterogeneous systems with multi-rate data dependencies and rate intervals. Technical report, IDA, TU-Braunschweig, Dec. 2002.

[8] E. A. Lee and D. G. Messerschmitt. Synchronous dataflow. *Proceedings of the IEEE*, 75(9):1235–1245, 1987.

[9] K. Richter and R. Ernst. Event model interfaces for heterogeneous system analysis. In *Proc. of Design, Automation and Test in Europe Conference (DATE'02)*, Paris, France, Mar. 2002.

[10] K. Richter, D. Ziegenbein, M. Jersak, and R. Ernst. Model composition for scheduling analysis in platform design. In *Proceeding 39th Design Automation Conference*, New Orleans, USA, June 2002.

[11] L. Thiele, S. Chakraborty, M. Gries, and S. Künzli. Design space exploration of network processor architectures. In *First Workshop on Network Processors at the 8th International Symposium on High-Performance Computer Architecture (HPCA8)*, Cambridge MA, USA, Feb. 2002.

[12] K. W. Tindell. An extendible approach for analysing fixed priority hard real-time systems. *Journal of Real-Time Systems*, 6(2):133–152, Mar 1994.

[13] Y. Xie and W. Wolf. Allocation and scheduling of conditional task graph in hardware/software co-synthesis. In *Proc. Design, Automation and Test in Europe (DATE'01)*, Munich, Germany, Mar. 2001.

[14] D. Ziegenbein, M. Jersak, K. Richter, and R. Ernst. Breaking down complexity for reliable system-level timing validation. In *Ninth IEEE/DATC Electronic Design Processes Workshop (EDP'02)*, Monterey, USA, Apr. 2002.

[15] D. Ziegenbein, J. Uerpmann, and R. Ernst. Dynamic Response Time Optimization for SDF Graphs. In *Proceedings International Conference on Computer-Aided Design (ICCAD '00)*, San Jose, Nov. 2000.