ATS21

Document version: 0.4
ATS21 is a programmable multi-clock/timer/alarm component, created in 2021.

This design is imaginary but intended to reflect some aspects of a real design, which for pragmatic, commercial reasons may cause the design to appear to have "imperfect" or awkwardly defined features (perhaps due to time constraints, backwards-compatibility requirements, or first use limitations). The design will be dynamically created as part of an exercise where students play the role of a contributor to the early definition of an architectural feature. As this is a verification class, your role is through the lens of a verification engineer, though part of that role (particularly at this stage of a project) is to also be part architect, part designer, and part project manager.

Assignment 1: Create a reference design that can be used to check whether an implementation of the design is correct. Note the implementation could be another student team's design. A significant aspect of this exercise is your participation in the creation of the specification of the design. This design will be used in subsequent assignments.

As a commercial product definition is being formed, verification engineers (at least some) should be participating in the conversation (with their verification "hats" on). This helps a project in several ways. The design is scrutinized early to prevent bugs before code is written and to be sure that the definition is complete and precise. Project also need to adhere to schedules so verification experts need to consider whether the verification of a feature can be accomplished in the time available. If not, the product definition will need to be reduced.

A result of this process should be a specification document that can be used by design and verification. Participants also gain a richer understanding of the design and can act as experts later in the project to ensure development progresses smoothly.

Assignment 2 will require that a coverage model be constructed (this will be explained in week 2). Coverage data should then be collected and analyzed using totally (or nearly totally) random input stimulus. For assignment 3, testbenches will add constrained input stimulus and the resulting coverage collected and analyzed.

ATS21 Overview

Modern ASIC designs often integrate many individually design components or IPs into an SOC IC. Many IPs will need to rely upon a common notion of clocking, potentially synchronized with other devices. The ATS21 supports management of up to 16 16-bit programmable base clocks and up to 24 independent 16-bit programmable alarms/timers. The design also supports concurrent requests from two clients for updates to the ATS21 behavior.

Base Clocks:

Each base clock (BC) can be set to any 16-bit value and increments synchronously relative to the clock input (**clk**) to the ATS21. Each BC also has an attribute to indicate how frequently it increments. A BC may increment with

every **clk** (full speed- rate: 00),
every other **clk** (half speed – rate: 01),
or every fourth **clk** (quarter speed - rate: 10).

Each BC can also be programmatically enabled or disabled (paused).

Alarm/Timers:

Each Alarm/Timer (AT) can be programmed to assert on an event relative to one of the base clocks.

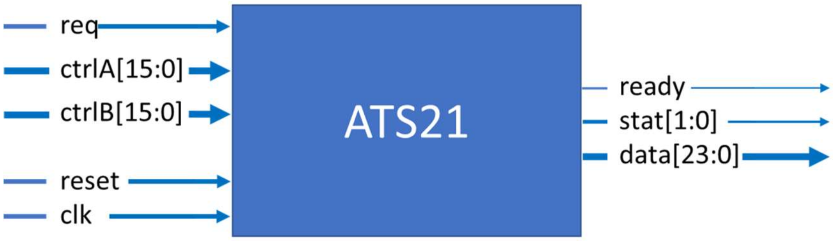
- The event can indicate either a specific BC value (time) has been reached (alarm) or some number of intervals has passed (timer) since the AT was enabled.
- When an event has been reached, an output pin corresponding to the AT will be asserted (**data[]**) for 2 **clk** clock cycles.
- Any AT can be programmatically enabled or disabled (paused).

- While an AT used as a timer generates a one-time event, if the AT is used as an alarm it may be programmed to cause multiple events each time the alarm value is set.
- When set, base clocks run at full speed and are enabled immediately (not paused).
- When set, ATs run are enabled immediately (not paused).
- When an AT (alarm or timer) is set, it replaces/overrides any preexisting data for the identified AT.

Dual-Ported control: 16-bit ports are provided (**ctrlA** and **ctrlB**) so that two requests may be submitted simultaneously to the ATS21. Requests are in the form of 32-bit instructions that make modifications to any of the BCs or ATs. There is also one control register (discussed below) that enables or disables requests. Since the ports are only 16-bits, but instructions require 32 bits, instructions are submitted in 16-bit chunks in two consecutive clock cycles.

ATS21 Control Register: The control register enables or disables ATS activities. The device can be activated/inactivated and each of the ports can be enable or disable to accept either instructions to set BC values or AT values.

Block Diagram



Pins

Inputs

req	Request to device being made
clk	All other signals are synchronous with the clock
reset	When asserted for at least 2 cycles, device is reset and all registers cleared. This input is synchronous with the clk input.
ctrlA[16]	Request from Client A – two cycles are required to input 32-bit operation
ctrlB[16]	Request from Client B – two cycles are required to input 32-bit operation

Outputs

ready	ATS21 is ready to accept new operation requests. This signal may fall during reset an rise when the device is fully reset.
stat[2]	stat[0] provides a response to a previous request for Client A stat[1] provides a response to a previous request for Client B If a request is accepted (ack=1), the stat output signal should be asserted. The stat value should remain low (nack=0) if there was no request or if there was an error (the request was not accepted).
data[24]	Bits indicating one of the 24 possible alarms/timers has been

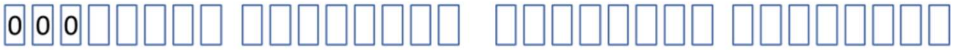
Registers

CR bits

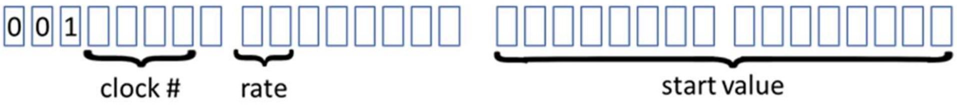
0	Device inactive/active
1	Timer/Alarm change allowed by Client A
2	Timer/Alarm change allowed by Client B
3	Clock change allowed by Client A
4	Clock change allowed by Client B
5-7	Unused

Operations

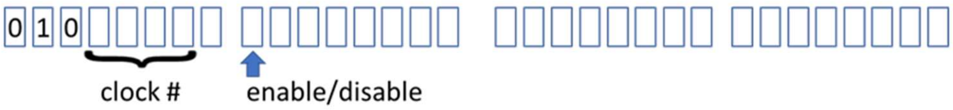
no operation



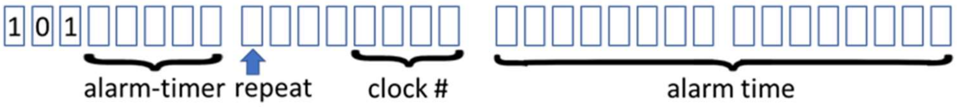
set clock



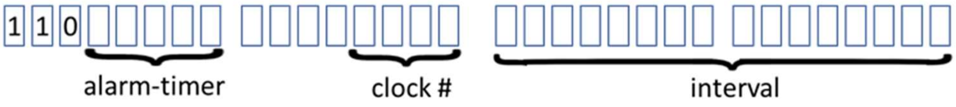
enable or disable clock



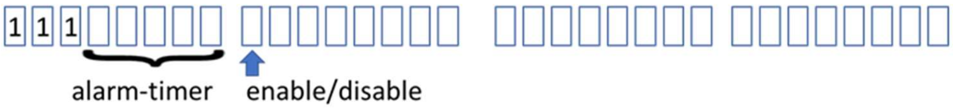
set alarm



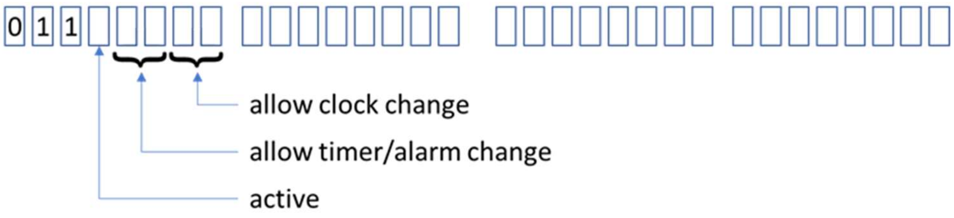
set countdown timer



enable or disable alarm/timer



set ATS21 mode



Miscellaneous Details

- Your reference model design does not need to be synthesizable.

- **Requests**
  - **req** is asserted (=1) if a new request by either ClientA or ClientB is initiated. The value only needs to be asserted during the first cycle of driving a request into the design.
  - During the first cycle the top 16 bits that includes the opcode is driven into the device.
  - During the second cycle the bottom 16 bits are driven into the device.
  - Since **req** may be asserted for either ClientA or ClientB, if a client is not making a request, its top 3 ctrl pins should be 0 (noop).
  - During the second cycle of a request being driven, if **req** is high, it would indicate that the other Client may be initiating a request.

- Frequency changes may occur at any time without first pausing a base clock or AT.

- The precise behavior of **ready** (when it is asserted and deasserted) is implementation dependent and does not need to be checked at this time.