2023 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION UNITED STATES

SAN JOSE, CA, USA
FEBRUARY 27-MARCH 2, 2023

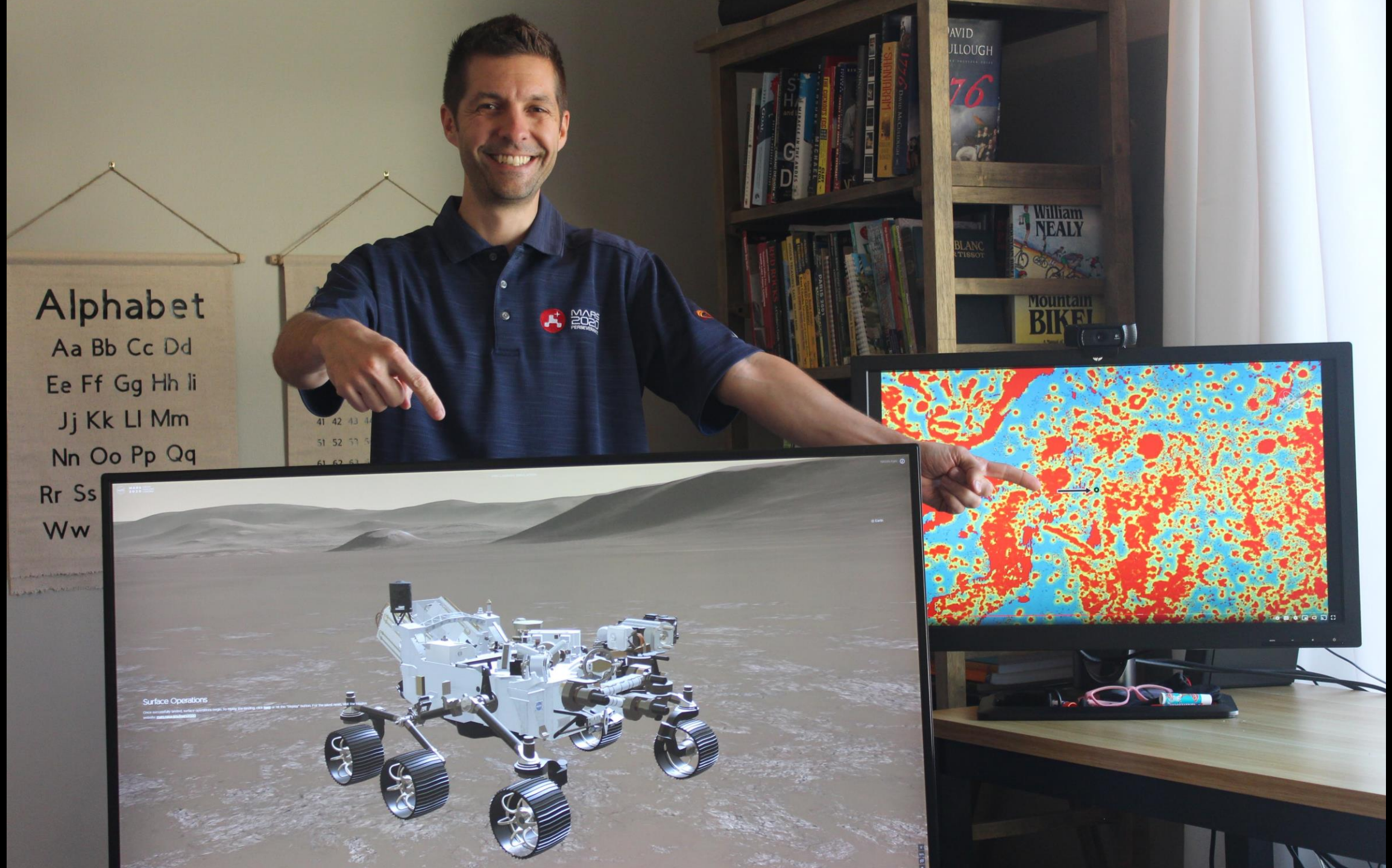
Regvue
Modern Hardware/Software Interface Documentation

Rob Donnelly NASA-JPL

Josh Geden NASA-JPL

Jet Propulsion Laboratory
California Institute of Technology

accellera SYSTEMS INITIATIVE


Alphabet
Aa Bb Cc Dd
Ee Ff Gg Hh Ii
Jj Kk Ll Mm
Nn Oo Pp Qq
Rr Ss
Ww
Surface Operations
William NEALY
Mountain BIKE

A Silly Debug Story

- Manual test of error logic in physical hardware
- Cannot clear the error status bits
- Multiple engineers stumped
- Root cause: mistyped/misread data values
  - E.g. Writing `0x0001000` instead of `0x00010000`
  - E.g. Writing `00010030` instead of `0x00010030`

A Need Presents

- Humans are error prone
  - Decoding **register** values into **field** values
  - Encoding **field** values into **register** values
  - Decoding **numeric** values into **symbolic** values (enumerations)
  - Byte swapping
- Inefficient information sharing
- Inefficient information access

A Solution Develops

- Modern web technologies
- Static + ***interactive*** content
- Direct permalinks
- Client-side web application
  - No client install required
  - Simple hosting

regvue - SiFive FE310 RISC-V SoC

github.jpl.nasa.gov/pages/regvue/regvue/v1/#/root/uart0/txdata?field=data&data=https://github.jpl.nasa.gov/regvue/demos/raw/main/od...

SiFive FE310 RISC-V SoC v1.0 | Search Ctrl+K | SiFive GitHub | Press Release

clint	0x2000000
plic	0xc000000
wdog	0x10000000
rtc	0x10000000
aonclk	0x10000000
backup	0x10000000
pmu	0x10000000
prci	0x10008000
otp	0x10010000
gpio0	0x10012000
uart0	0x10013000
txdata	0x10013000
rxdata	0x10013004
txctrl	0x10013008
rxctrl	0x1001300c
ie	0x10013010
ip	0x10013014
div	0x10013018
div_enum	0x10013018
qspi0	0x10014000
pwm0	0x10015000
i2c0	0x10016000

Transmit Data FIFO Register - uart0.txdata

0x10013000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
full	rsvd0																							data							
0	0x000000																							0x00							
0x00000000																															

Base: B D **H** | Swap: Byte Word | Reset: POR

Writes to this register push data into the UART transmit FIFO.

Bits	Name	Access	Description
31	full	ro	Indicates whether the transmit FIFO is full. If this bit is 1, writes to this register will be dropped. This bit is updated on write.
30:8	rsvd0	rsvd	Reserved
7:0	data	rw	For writes, the data to push into the FIFO. For reads, the data that was last pushed into the FIFO.

Powered by regvue v1.1.0

2023
DESIGN AND VERIFICATION™
DVCON
CONFERENCE AND EXHIBITION
UNITED STATES
SAN JOSE, CA, USA
FEBRUARY 27-MARCH 2, 2023

Documentation Comparison

accellera
SYSTEMS INITIATIVE

PDF Register Example

24.6.1 SPI control register 1 (SPIx_CR1)

Address offset: 0x00

Reset value: 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BIDI MODE	BIDI OE	CRC EN	CRC NEXT	CRCL	RX ONLY	SSM	SSI	LSB FIRST	SPE	BR [2:0]			MSTR	CPOL	CPHA
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bit 15 **BIDIMODE:** Bidirectional data mode enable. This bit enables half-duplex communication using common single bidirectional data line. Keep RXONLY bit clear when bidirectional mode is active.

0: 2-line unidirectional data mode selected

1: 1-line bidirectional data mode selected

Bit 14 **BIDIOE:** Output enable in bidirectional mode

This bit combined with the BIDIMODE bit selects the direction of transfer in bidirectional mode

0: Output disabled (receive-only mode)

1: Output enabled (transmit-only mode)

Note: In master mode, the MOSI pin is used and in slave mode, the MISO pin is used.

PDF Register Example

Limited space for 16 bits let alone 32 bits

Need to manually encode/decode field between field and register values

24.6.1 SPI control register 1 (SPIx_CR1)

Address offset: 0x00

Reset value: 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BIDI MODE	BIDI OE	CRC EN	CRC NEXT	CRCL	RX ONLY	SSM	SSI	LSB FIRST	SPE	BR [2:0]			MSTR	CPOL	CPHA
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bit 15 **BIDIMODE:** Bidirectional data mode enable. This bit enables half-duplex communication using common single bidirectional data line. Keep RXONLY bit clear when bidirectional mode is active.

0: 2-line unidirectional data mode selected

1: 1-line bidirectional data mode selected

Bit 14 **BIDIOE:** Output enable in bidirectional mode

This bit combined with the BIDIMODE bit selects the direction of transfer in bidirectional mode

0: Output disabled (receive-only mode)

1: Output enabled (transmit-only mode)

Note: In master mode, the MOSI pin is used and in slave mode, the MISO pin is used.

Limited to short, cryptic identifiers (presentation > meaning)

Missing reset value

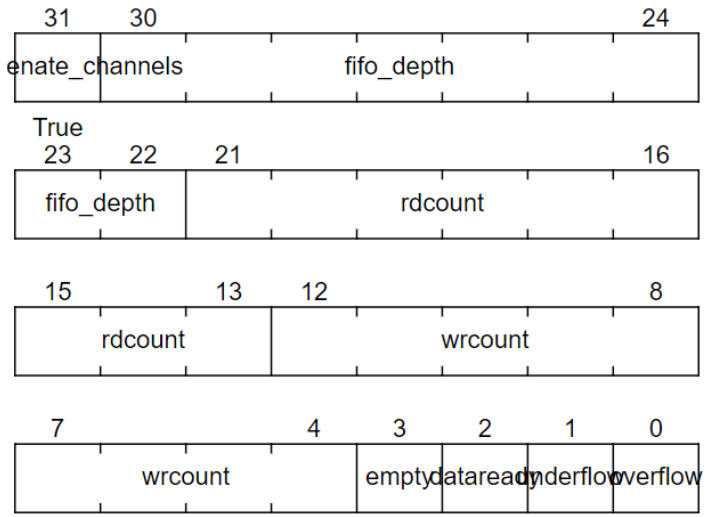
Missing software access type

HTML Register Example

AUDIO_RX_STAT

Address: 0xf001b000 + 0x10 = 0xf001b010

Rx data path status



Field	Name	Description
[0]	OVERFLOW	Rx overflow
[1]	UNDERFLOW	Rx underflow
[2]	DATAREADY	256 words of data loaded and ready to read
[3]	EMPTY	No data available in FIFO to read
[12:4]	WRCOUNT	Write count
[21:13]	RDCOUNT	Read count
[30:22]	FIFO_DEPTH	FIFO depth as synthesized
[31]	CONCATENATE_CHANNELS	Receive and send both channels atomically

HTML Register Example

AUDIO_RX_STAT

Address: 0xf001b000 + 0x10 = 0xf

Rx data path status

31	30	24
enate_channels	fifo_depth	

True

23	22	21	16
fifo_depth	rdcount		

15	13	12	8
rdcount	wrcount		

7	4	3	2	1	0
wrcount		empty	dataready	underflow	overflow

Text overflow and truncation (meaning > presentation)

Missing software access type

Multiple rows

Text overflow

	Name	Description
[0]	OVERFLOW	Rx overflow
[1]	UNDERFLOW	Rx underflow
[2]	DATAREADY	256 words of data loaded and ready to read
[3]	EMPTY	No data available in FIFO to read
[12:4]	WRCOUNT	Write count
[21:13]	RDCOUNT	Read count
[30:22]	FIFO_DEPTH	FIFO depth as synthesized
[31]	CONCATENATE_CHANNELS	Receive and send both channels atomically

Regvue Register Example

rx_stat - audio.rx_stat

0xf0019010

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
concatenate_channels	fifo_depth									rdcount									wrcount									empty	dataready	underflow	overflow
1	0x000									0x000									0x000									0	0	0	0
0x80000000																															

Base B D H Swap Byte Word Reset

Bits ⇅	Name	Access	Description
31	concatenate_channels	rw	Receive and send both channels atomically
30:22	fifo_depth	ro	FIFO depth as synthesized
21:13	rdcount	w1c	Read count
12:4	wrcount	w1c	Write count
3	empty	ro	No data available in FIFO to read
2	dataready	ro	256 words of data loaded and ready to read
1	underflow	w1c	Rx underflow
0	overflow	w1c	Rx overflow

Regvue Register Example

Arbitrary length identifiers

Full symbolic path

Sortable

Software access type

rx_stat - audio.rx_stat

0xf0019010

31	30:22	21:13	12:4	3	2	1	0
concatenate_channels	fifo_depth	rdcount	wrcount	empty	dataread	underflo	overflow
1	0x000	0x000	0x000	0	0	0	0
0x80000000							

Base: B D H Swap: Byte Word

Mouseover highlighting

Field values

Register value

Bits	Name	Access	Description
31	concatenate_channels	rw	Receive and send both channels atomically
30:22	fifo_depth	ro	FIFO depth as synthesized
21:13	rdcount	w1c	Read count
12:4	wrcount	w1c	Write count
3	empty	ro	No data available in FIFO to read
2	dataready	ro	256 words of data loaded and ready to read
1	underflow	w1c	Rx underflow
0	overflow	w1c	Rx overflow

PDF Address Map Example

Bus	Boundary address	Size	Peripheral	Peripheral register map
APB	0x4001 5C00 - 0x4001 7FFF	9 KB	Reserved	-
	0x4001 5800 - 0x4001 5BFF	1 KB	DBGMCU	-
	0x4001 4C00 - 0x4001 57FF	3 KB	Reserved	-
	0x4001 4800 - 0x4001 4BFF	1 KB	TIM17	*Section 17.6.16 on page 468*
	0x4001 4400 - 0x4001 47FF	1 KB	TIM16	*Section 17.6.16 on page 468*
	0x4001 4000 - 0x4001 43FF	1 KB	TIM15	*Section 17.5.18 on page 449*
	0x4001 3C00 - 0x4001 3FFF	1 KB	Reserved	-
	0x4001 3800 - 0x4001 3BFF	1 KB	USART1	*Section 23.7.11 on page 640*
	0x4001 3400 - 0x4001 37FF	1 KB	Reserved	-
	0x4001 3000 - 0x4001 33FF	1 KB	SPI1	*Section 24.6.8 on page 676*
	0x4001 2C00 - 0x4001 2FFF	1 KB	TIM1	*Section 13.4.21 on page 294*
	0x4001 2800 - 0x4001 2BFF	1 KB	Reserved	-
	0x4001 2400 - 0x4001 27FF	1 KB	ADC	*Section 12.11.11 on page 220*
	0x4001 1800 - 0x4001 23FF	4 KB	Reserved	-
	0x4001 1400 - 0x4001 17FF	1 KB	USART6	*Section 23.7.11 on page 640*
	0x4001 0800 - 0x4001 23FF	7 KB	Reserved	-
	0x4001 0400 - 0x4001 07FF	1 KB	EXTI	*Section 11.3.7 on page 180*
	0x4001 0000 - 0x4001 03FF	1 KB	SYSCFG	*Section 9.1.7 on page 151*
-	0x4000 8000 - 0x4000 FFFF	32 KB	Reserved	-

PDF Address Map Example

Bus	Boundary address	Size	Peripheral	Peripheral register map
APB	0x4001 5C00 - 0x4001 7FFF	9 KB	Reserved	-
	0x4001 5800 - 0x4001 5BFF	1 KB	DBGMCU	-
	0x4001 4C00 - 0x4001 57FF	3 KB	Reserved	-
	0x4001 4800 - 0x4001 4BFF	1 KB	TIM17	*Section 17.6.16 on page 468*
	0x4001 4400 - 0x4001 47FF	1 KB	TIM16	*Section 17.6.16 on page 468*
	0x4001 4000 - 0x4001 43FF	1 KB	TIM15	*Section 17.5.18 on page 449*
	0x4001 3C00 - 0x4001 3FFF	1 KB	Reserved	-
	0x4001 3800 - 0x4001 3BFF	1 KB	USART1	*Section 23.7.11 on page 640*
	0x4001 3400 - 0x4001 37FF	1 KB	Reserved	-
	0x4001 3000 - 0x4001 33FF	1 KB	SPI1	*Section 24.6.8 on page 676*
	0x4001 2C00 - 0x4001 2FFF	1 KB	TIM1	*Section 13.4.21 on page 294*
	0x4001 2800 - 0x4001 2BFF	1 KB	Reserved	-
	0x4001 2400 - 0x4001 27FF	1 KB	ADC	*Section 12.11.11 on page 220*
	0x4001 1800 - 0x4001 23FF	4 KB	Reserved	-
	0x4001 1400 - 0x4001 17FF	1 KB	USART6	*Section 23.7.11 on page 640*
	0x4001 0800 - 0x4001 23FF	7 KB	Reserved	-
	0x4001 0400 - 0x4001 07FF	1 KB	EXTI	*Section 11.3.7 on page 180*
	0x4001 0000 - 0x4001 03FF	1 KB	SYSCFG	*Section 9.1.7 on page 151*
-	0x4000 8000 - 0x4000 FFFF	32 KB	Reserved	-

Verbose

Additional details elsewhere

Regvue Address Map Example

pwr	0x40007000
i2c1	0x40005400
i2c2	0x40005800
iwdg	0x40003000
wwdg	0x40002c00
tim1	0x40012c00
tim3	0x40000400
tim14	0x40002000
tim6	0x40001000
exti	0x40010400
nvic	0xe000e100
dma	0x40020000
rcc	0x40021000
syscfg	0x40010000
adc	0x40012400
isr	0x40012400
ier	0x40012404
cr	0x40012408
cfgr1	0x4001240c
cfgr2	0x40012410

cfgr1 - adc.cfgr1

0x4001240c

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
rsvd3	awdch					rsvd2		awden	awdsgl	rsvd1					discen	autoff	autdly	cont	ovrmod	exten		rsvd0	extsel			align	res		scandir	dmacfg	dmaen
0	0x02					0x0		0	0	0x00					0	1	0	1	0	0x0		0	0x0			0	0x0		0	0	0
0x0800A000																															

Base: B D **H** Swap: Byte Word Reset

configuration register 1

Bits	Name	Access	Description
31	rsvd3	rsvd	Reserved
30:26	awdch	rw	Analog watchdog channel selection
25:24	rsvd2	rsvd	Reserved

Regvue Address Map Example

Simple address map

Hierarchical

Map and register side-by-side

pwr	07000
i2c1	05400
i2c2	0x40005800
iwdg	0x40003000
wwdg	0x40002c00
tim1	0x40012c00
tim3	0x40000400
tim14	0x40002000
tim6	0x40001000
exti	
rcc	0x40021000
syscfg	0x40010000
adc	0x40012400
isr	0x40012400
ier	0x40012404
cr	0x40012408
cfgr1	0x4001240c
cfgr2	0x40012410

cfgr1 - adc.cfgr1

0x4001240c

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
rsvd3	awdch					rsvd2		awden	awdsgl	rsvd1					discen	autoff	autdly	cont	ovrmod	exten		rsvd0	extsel			align	res		scandir	dmacfg	dmaen
0	0x02					0x0		0	0	0x00					0	1	0	1	0	0x0		0	0x0			0	0x0		0	0	0
0x0800A000																															

Swap

Byte | Word

Reset

tion register 1

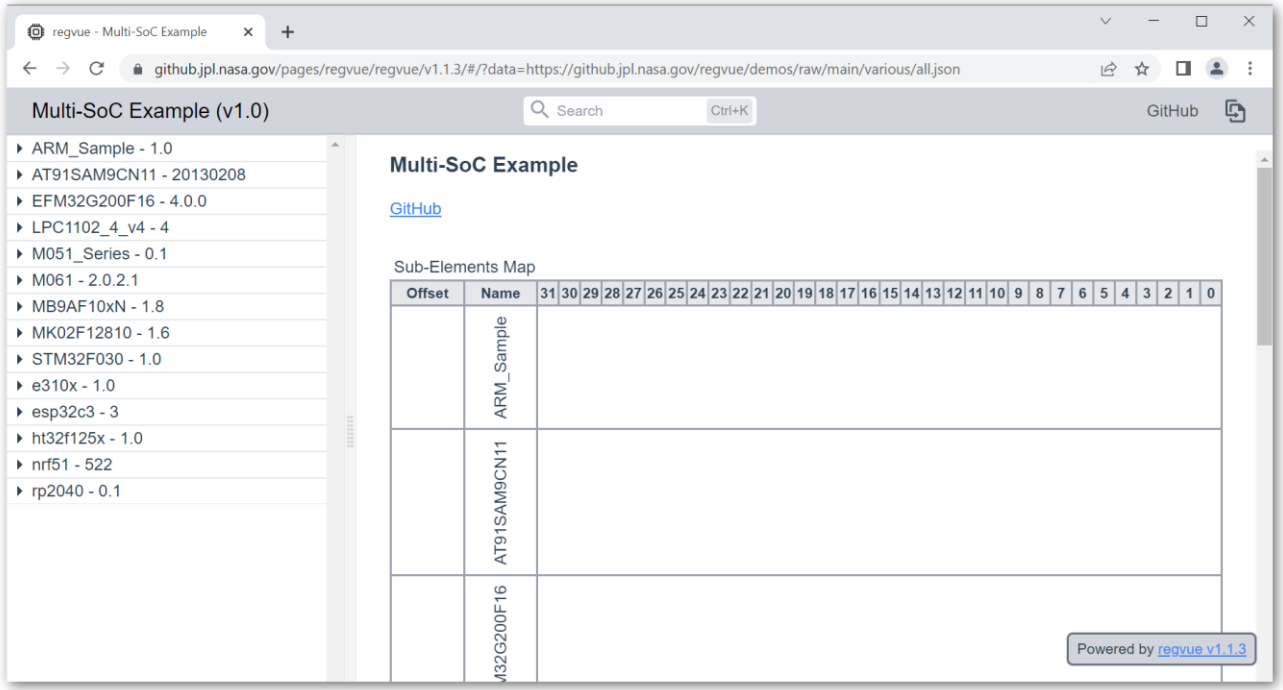
Bits	Name	Access	Description
31	rsvd3	rsvd	Reserved
30:26	awdch	rw	Analog watchdog channel selection
25:24	rsvd2	rsvd	Reserved

2023
DESIGN AND VERIFICATION™
DVCON
CONFERENCE AND EXHIBITION
UNITED STATES
SAN JOSE, CA, USA
FEBRUARY 27-MARCH 2, 2023

Feature Highlights

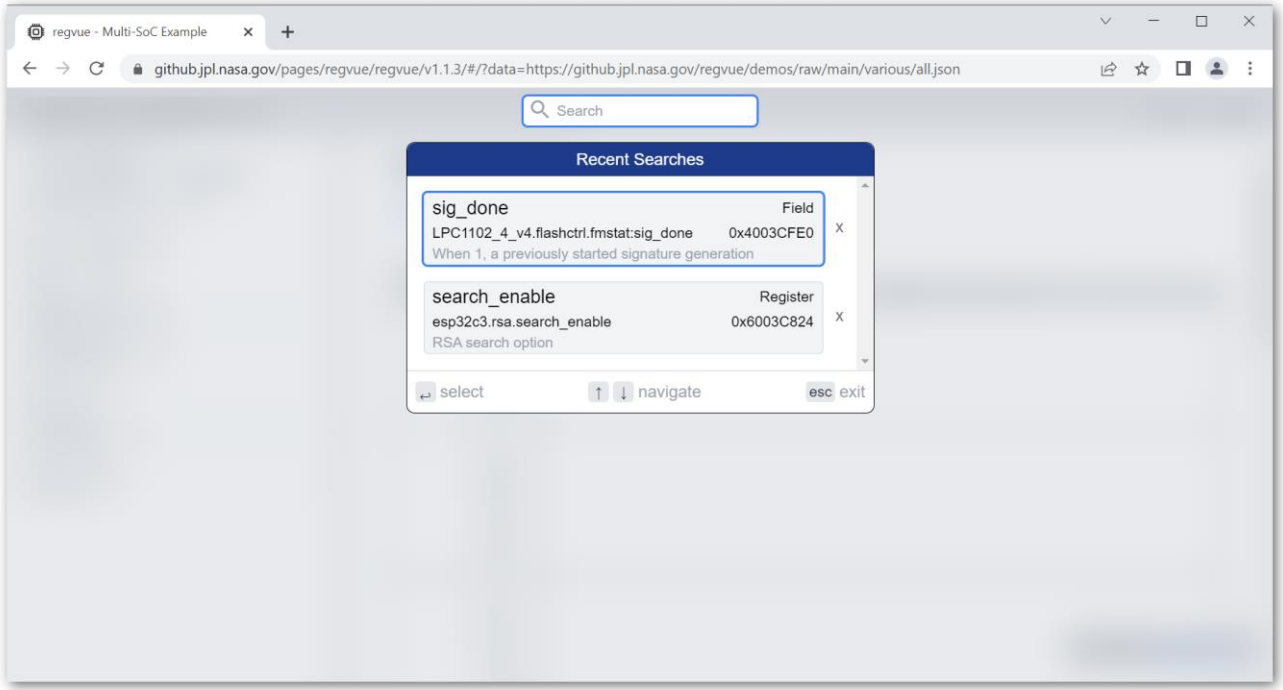
accellera
SYSTEMS INITIATIVE

Incremental Search



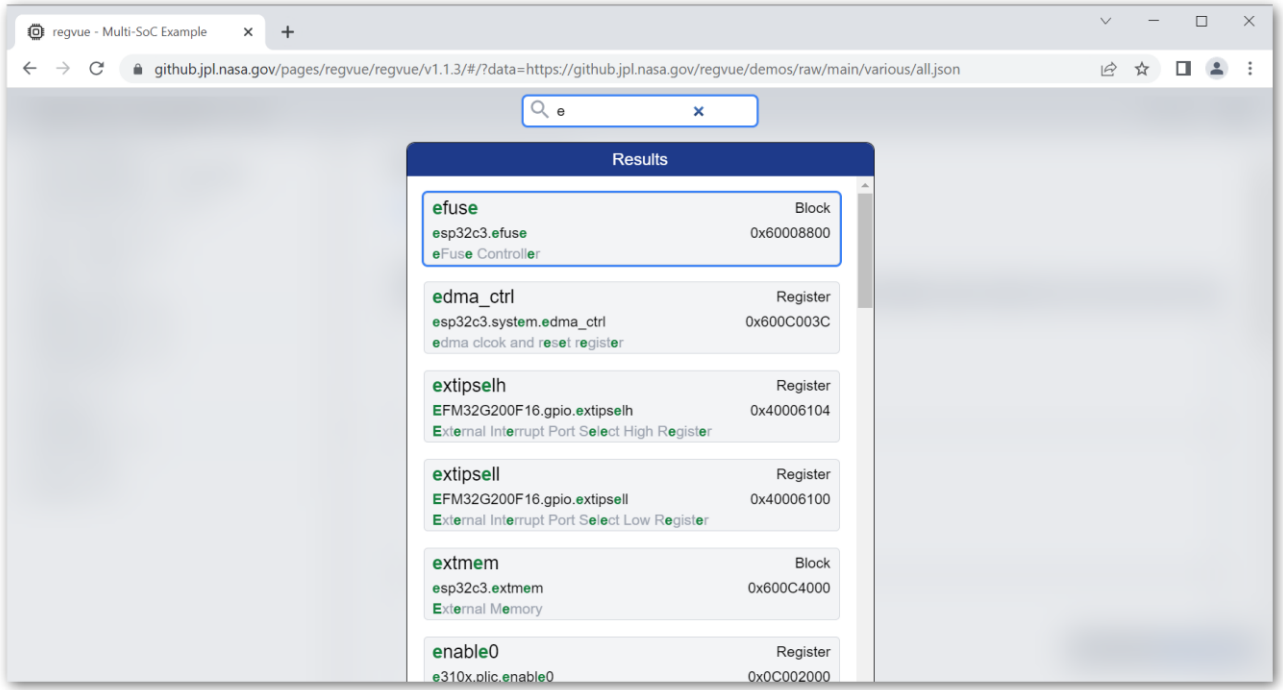
Incremental Search

- Press “Ctrl + k”



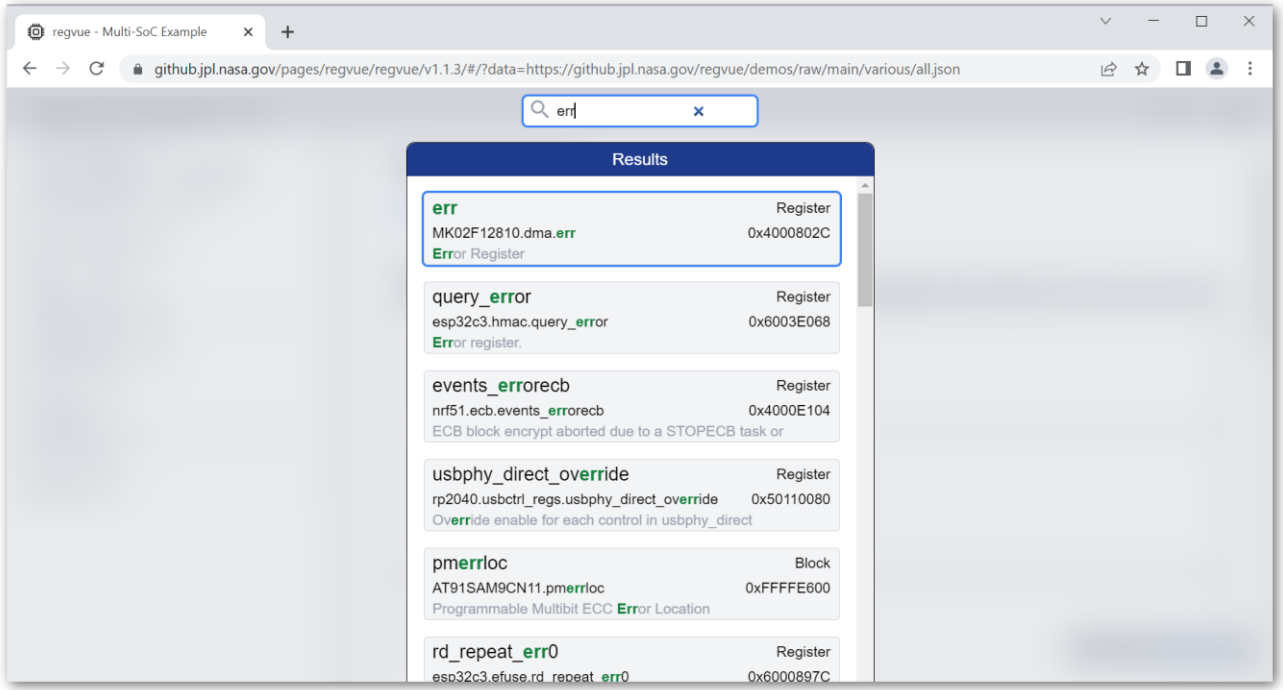
Incremental Search

- Press “Ctrl + k”
- Type “e”



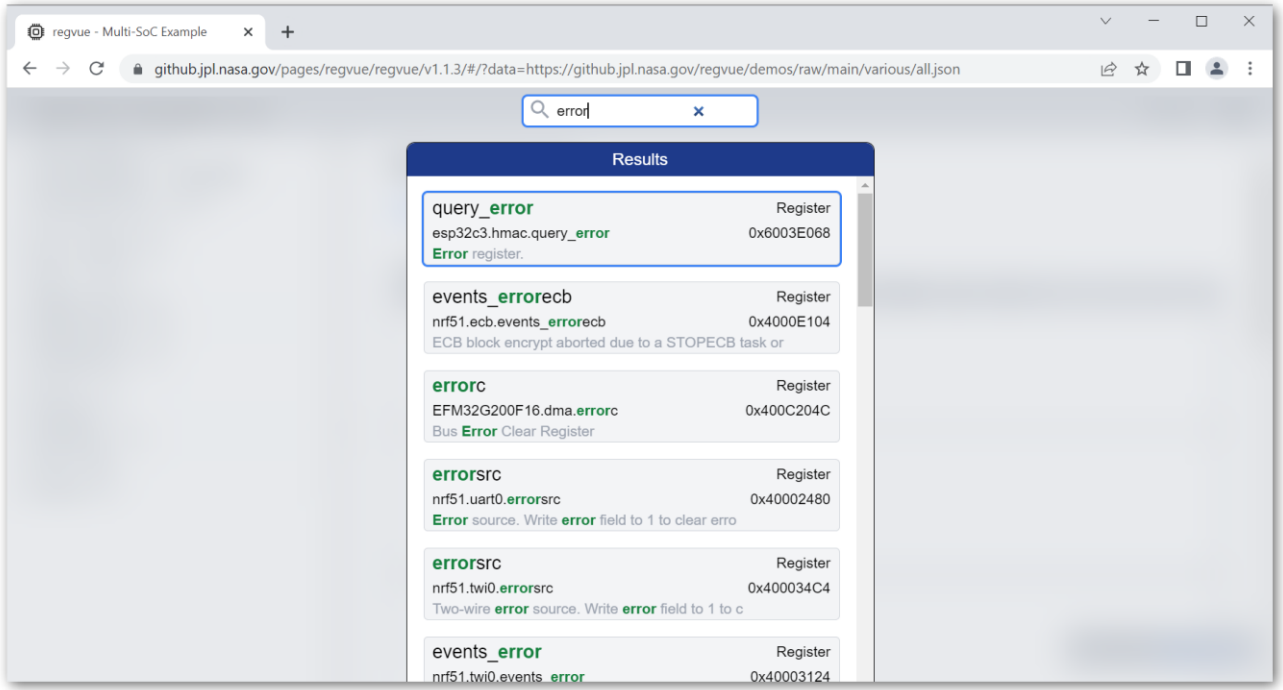
Incremental Search

- Press “Ctrl + k”
- Type “e”
- Type “rr”



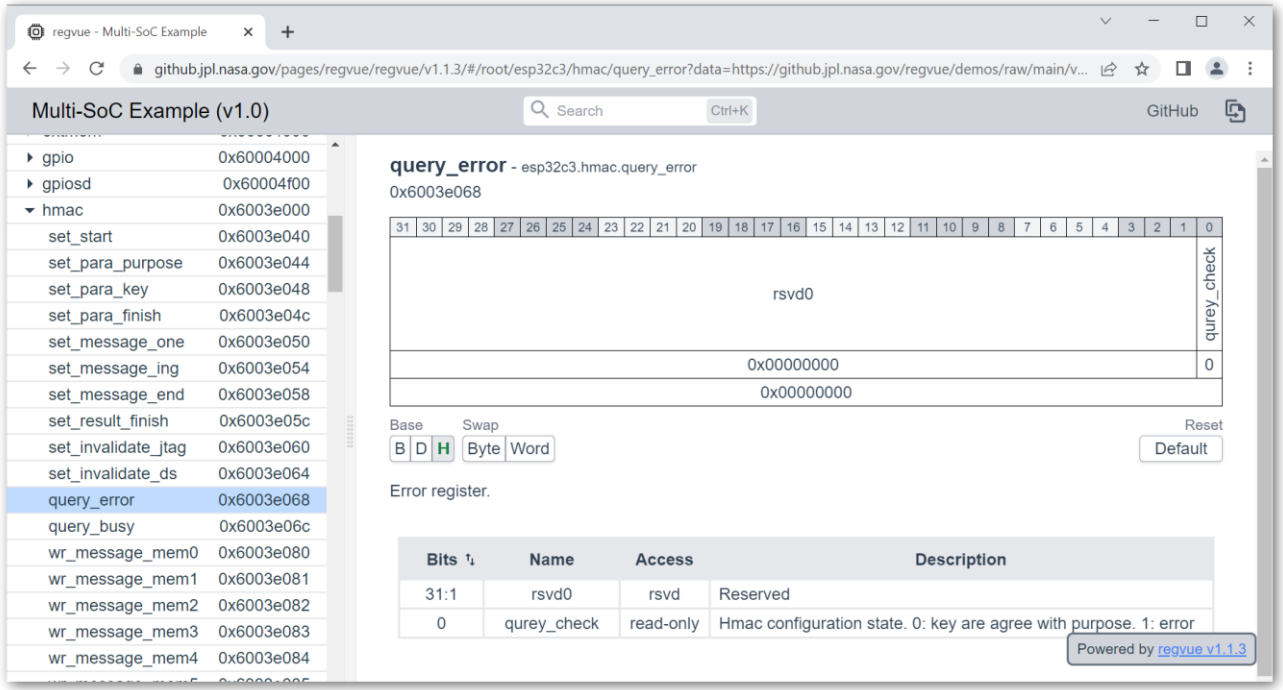
Incremental Search

- Press “Ctrl + k”
- Type “e”
- Type “rr”
- Type “or”



Incremental Search

- Press “Ctrl + k”
- Type “e”
- Type “rr”
- Type “or”
- Press “Enter”



Responsive Design

regvue - Multi-SoC Example

github.jpl.nasa.gov/pages/regvue/regvue/v1.1.3/#/root/ARM_Sample/timer0/cr?field=cntsrc&data=https://github.jpl.nasa.gov/regvue/demos/raw/main/various/all.json

Multi-SoC Example (v1.0)

Search Ctrl+K

GitHub

Horizontal field names

ARM_Sample - 1.0	
timer0	0x40010000
cr	0x40010000
sr	0x40010004
int	0x40010010
count	0x40010020
match	0x40010024
prescale_rd	0x40010028
prescale_wr	0x40010028
reload0	0x40010050
reload1	0x40010054
reload2	0x40010058
reload3	0x4001005c
timer1	0x40010100

cr - ARM_Sample.timer0.cr

0x40010000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
s	rsvd2			idr		reload		rsvd1		trgext		rsvd0		capedge		capsrc				cntsrc				psc	mode			cnt		rst	en
0	0x0			0x0		0x0		0x0		0x0		0x0		0x0		0x0				0x0				0	0x0			0x0		0	0
0x00000000																															

Base: B D H

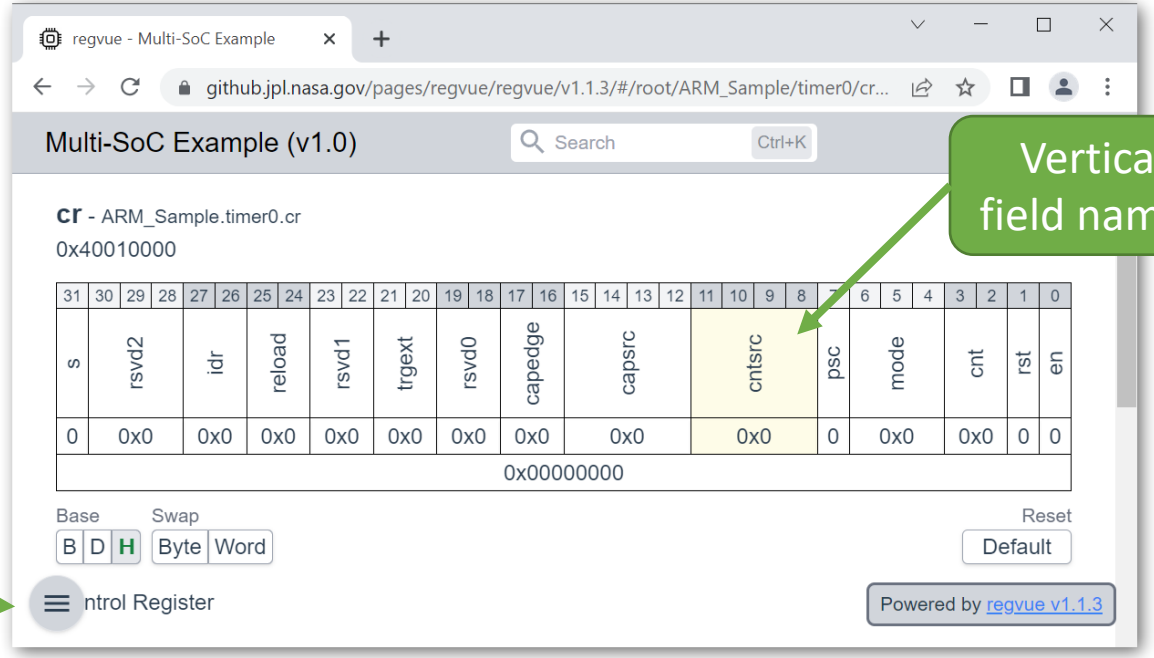
Swap: Byte Word

Reset: Default

Control Register

Powered by regvue v1.1.3

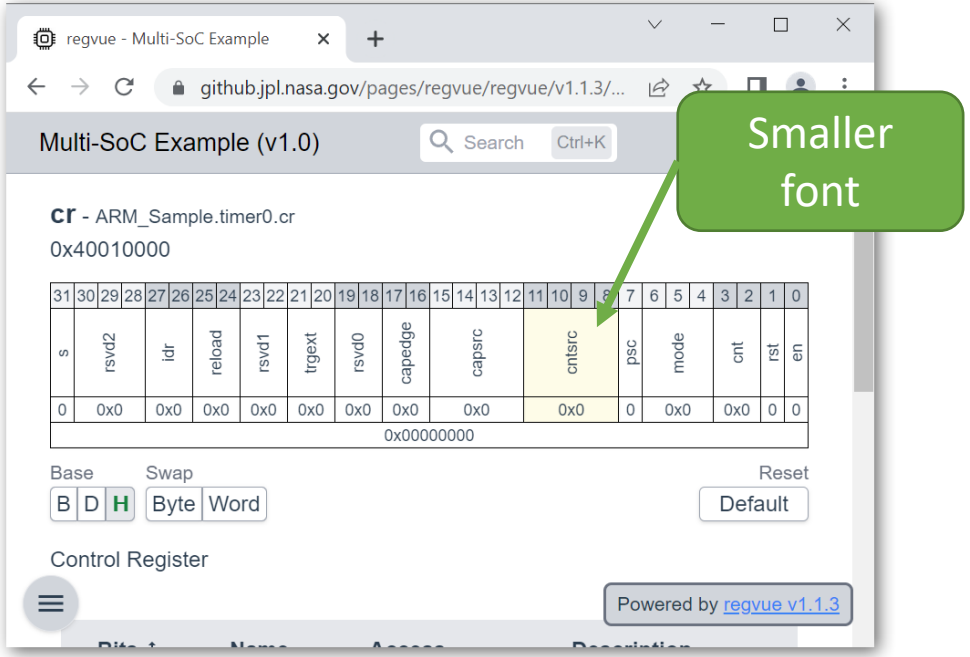
Responsive Design



Vertical field names

Address map hidden

Responsive Design



Encode/Decode

regvue - Multi-SoC Example

github.jpl.nasa.gov/pages/regvue/regvue/v1.1.3/#/root/ARM_Sample/timer0/cr?data=https://gith...

Multi-SoC Example (v1.0) Search Ctrl+K GitHub

cr - ARM_Sample.timer0.cr

0x40010000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
s	rsvd2			idr		reload		rsvd1		trgext		rsvd0		capedge		capsrc				cntsrc				psc	mode			cnt		rst	en
0	0x0			0x0		0x0		0x0		0x0		0x0		0x0		0x0				0x0				0	0x0			0x0		0	0
0x00000000																															

Base B D H Swap Byte Word Reset Default

Control Register

Bits	Name	Access	Description
31	s	read-write	Starts and Stops the Timer / Counter
30:28	rsvd2	rsvd	Reserved
27:26	idr	read-write	Selects, if Reload Register number is incremented, decremented or not modified
25:24	reload	read-write	Select RELOAD Register n to reload Timer on condition
23:22	rsvd1	rsvd	Reserved

Powered by regvue v1.1.3

Encode/Decode

- Select “psc” field value

regvue - Multi-SoC Example

github.jpl.nasa.gov/pages/regvue/regvue/v1.1.3/#/root/ARM_Sample/timer0/cr?data=https://gith...

Multi-SoC Example (v1.0) Search Ctrl+K GitHub

cr - ARM_Sample.timer0.cr

0x40010000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
s	rsvd2			idr		reload		rsvd1		trgext		rsvd0		capedge		capsrc				cntsrc				psc	mode			cnt		rst	en
0	0x0			0x0		0x0		0x0		0x0		0x0		0x0		0x0				0x0				0	0x0			0x0		0	0
0x00000000																															

Base B D H Swap Byte Word Reset Default

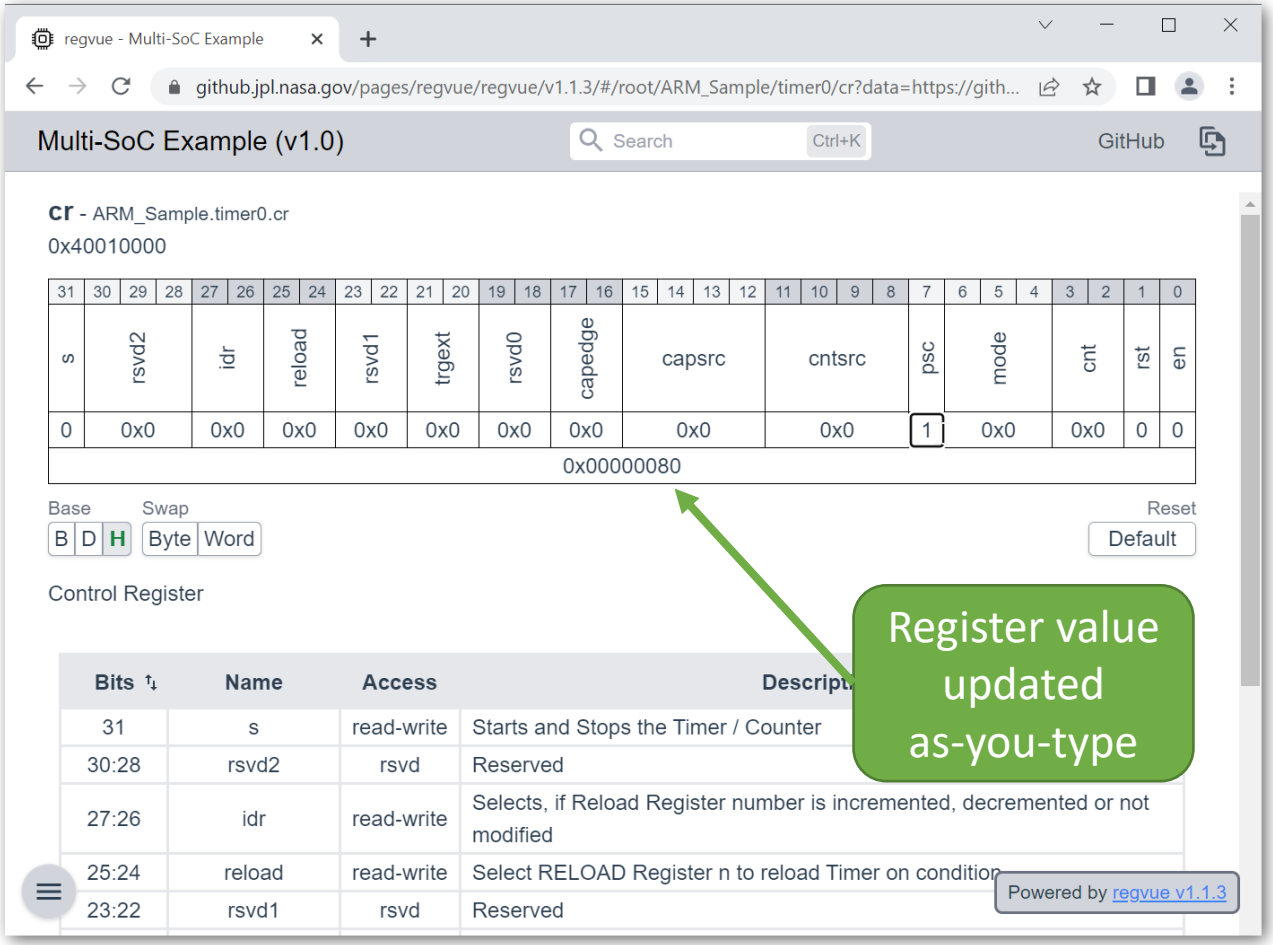
Control Register

Bits	Name	Access	Description
31	s	read-write	Starts and Stops the Timer / Counter
30:28	rsvd2	rsvd	Reserved
27:26	idr	read-write	Selects, if Reload Register number is incremented, decremented or not modified
25:24	reload	read-write	Select RELOAD Register n to reload Timer on condition
23:22	rsvd1	rsvd	Reserved

Powered by regvue v1.1.3

Encode/Decode

- Select “psc” field value
- Type “1”



Encode/Decode

- Select “psc” field value
- Type “1”
- Select register value

regvue - Multi-SoC Example

github.jpl.nasa.gov/pages/regvue/regvue/v1.1.3/#/root/ARM_Sample/timer0/cr?data=https://gith...

Multi-SoC Example (v1.0) Search Ctrl+K GitHub

cr - ARM_Sample.timer0.cr

0x40010000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
s	rsvd2			idr		reload		rsvd1		trgext		rsvd0		capedge		capsrc				cntsrc				psc	mode			cnt		rst	en
0	0x0			0x0		0x0		0x0		0x0		0x0		0x0		0x0				0x0				1	0x0			0x0		0	0

0x00000080

Base B D H Swap Byte Word Reset Default

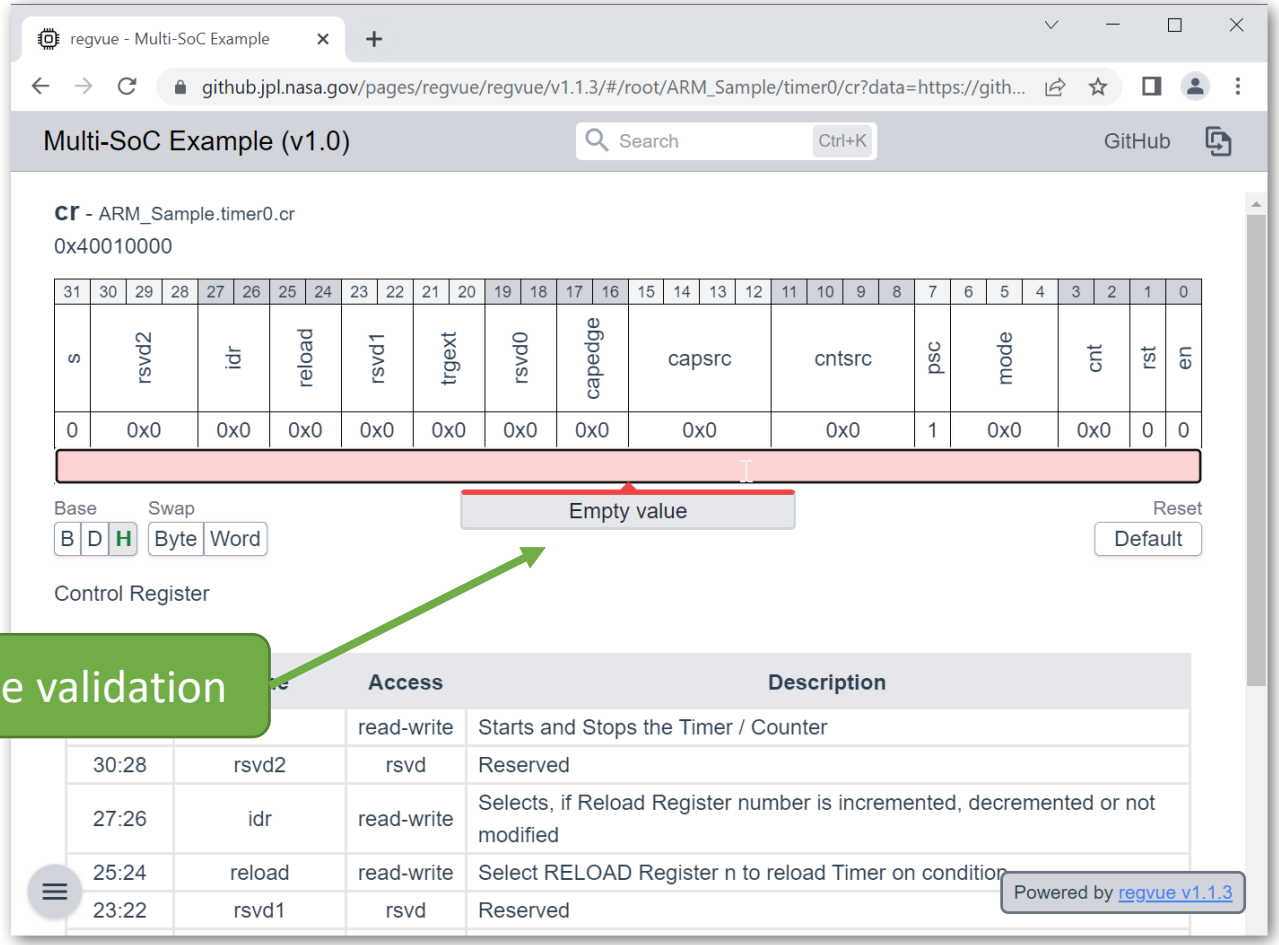
Control Register

Bits	Name	Access	Description
31	s	read-write	Starts and Stops the Timer / Counter
30:28	rsvd2	rsvd	Reserved
27:26	idr	read-write	Selects, if Reload Register number is incremented, decremented or not modified
25:24	reload	read-write	Select RELOAD Register n to reload Timer on condition
23:22	rsvd1	rsvd	Reserved

Powered by regvue v1.1.3

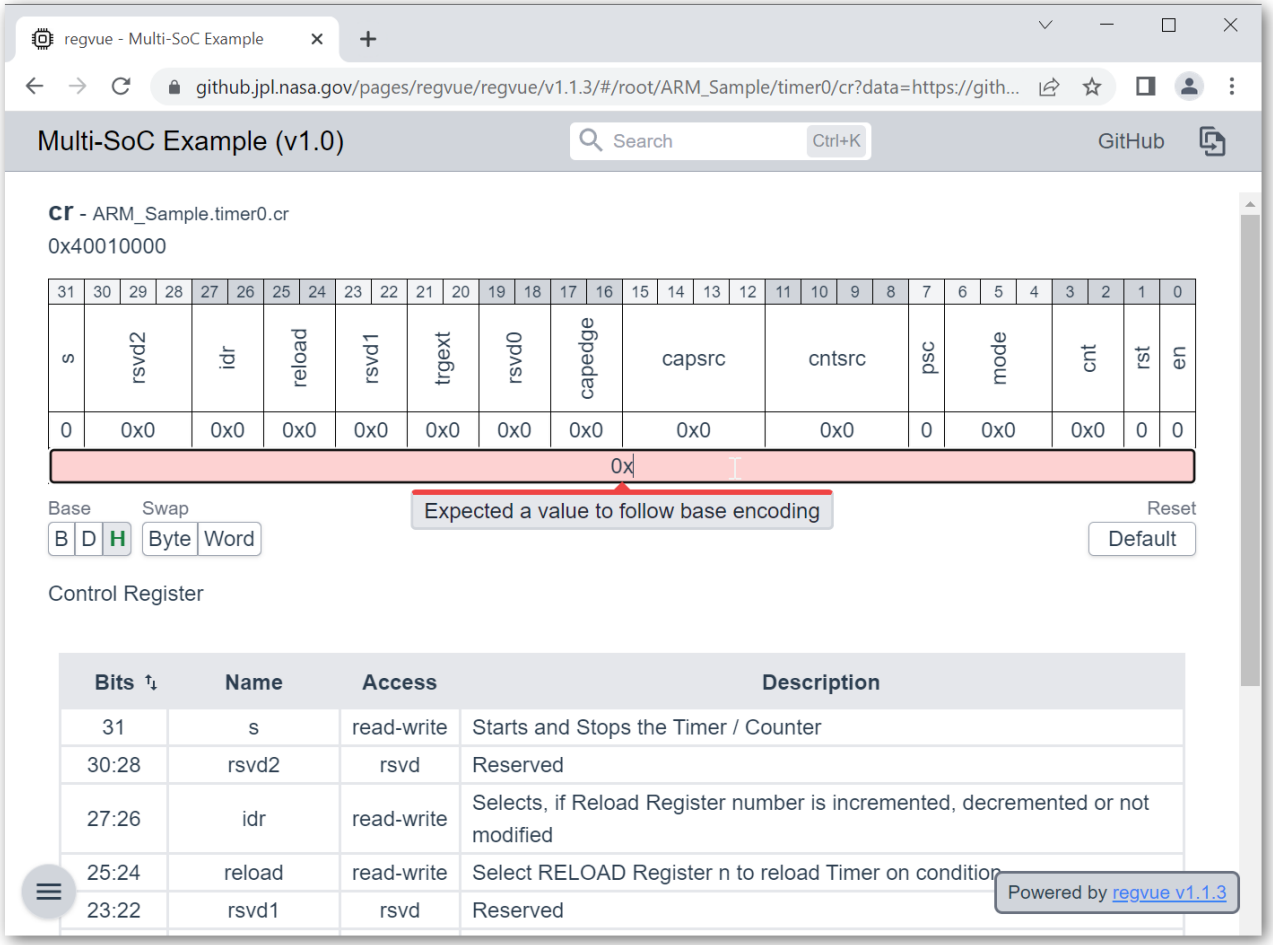
Encode/Decode

- Select “psc” field value
- Type “1”
- Select register value
- Press “Delete”



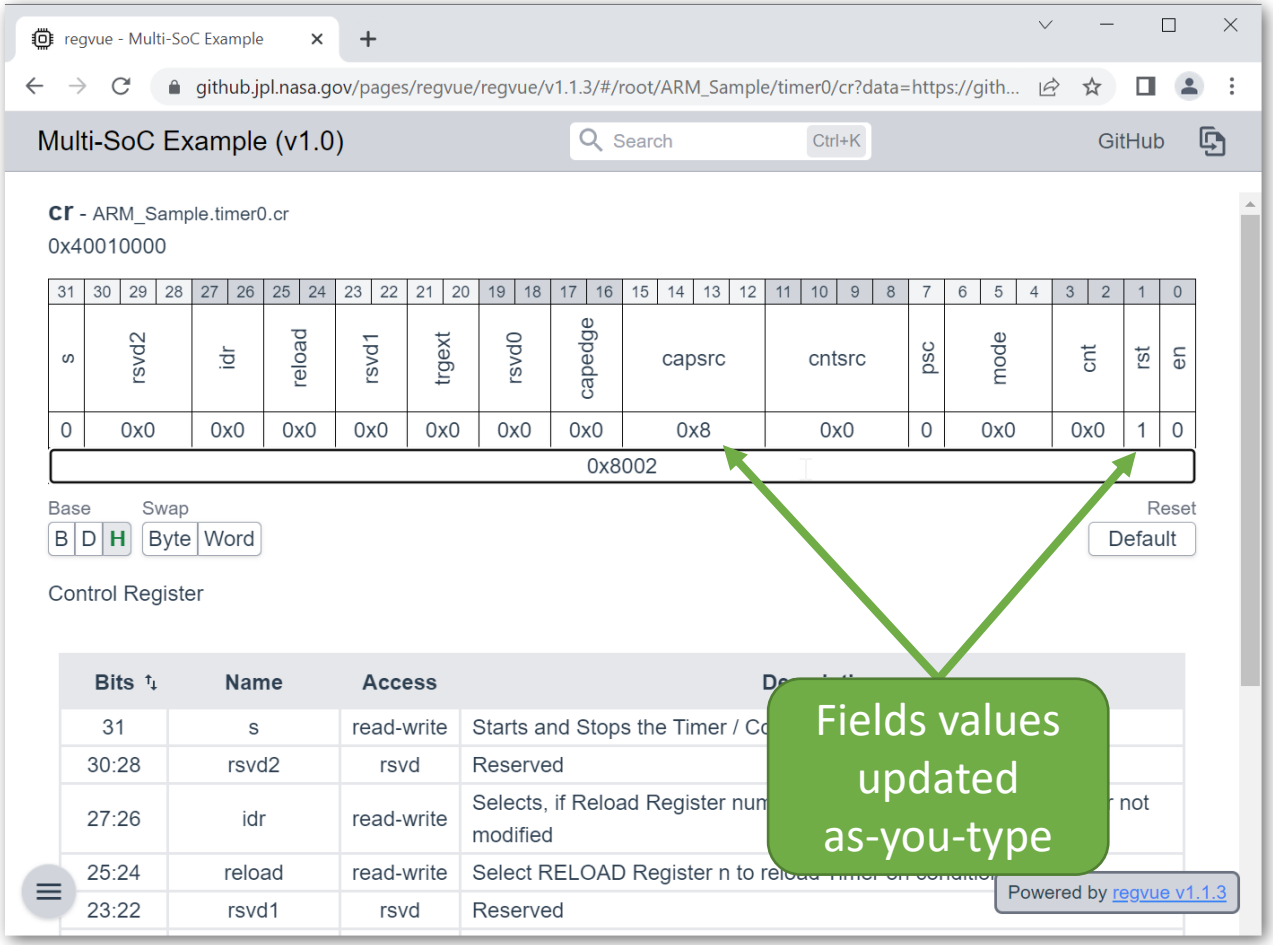
Encode/Decode

- Select “psc” field value
- Type “1”
- Select register value
- Press “Delete”
- Type “0x”



Encode/Decode

- Select “psc” field value
- Type “1”
- Select register value
- Press “Delete”
- Type “0x”
- Type “8002”



Encode/Decode

- Select “psc” field value
- Type “1”
- Select register value
- Press “Delete”
- Type “0x”
- Type “8002”
- Type “_abcd”

regvue - Multi-SoC Example

github.jpl.nasa.gov/pages/regvue/regvue/v1.1.3/#/root/ARM_Sample/timer0/cr?data=https://gith...

Multi-SoC Example (v1.0) Search Ctrl+K GitHub

cr - ARM_Sample.timer0.cr

0x40010000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
s	rsvd2			idr		reload		rsvd1		trgext		rsvd0		capedge		capsrc				cntsrc				psc	mode			cnt		rst	en
1	0x0			0x0		0x0		0x0		0x0		0x0		0x2		0xA				0xB				1	0x4			0x3		0	1
0x8002_abcd																															

Base: B D H Swap: Byte Word Reset: Default

Control Register

Bits	Name	Access	Description
31	s	read-write	Starts and Stops the Timer / Counter
30:28	rsvd2	rsvd	Reserved
27:26	idr	read-write	Selects, if Reload Register number is incremented, decremented or not modified
25:24	reload	read-write	Select RELOAD Register n to reload Timer on condition
23:22	rsvd1	rsvd	Reserved

Powered by regvue v1.1.3

Direct Permalinks

regvue - Multi-SoC Example

github.jpl.nasa.gov/pages/regvue/regvue/v1.1.3/#/root/ARM_Sample/timer0/cr?field=idr&data=https://github.jpl.nasa.gov/regvue/demos/raw/main/various/all.json

Multi-SoC Example (v1.0) | Search Ctrl+K | GitHub

- ARM_Sample - 1.0
  - timer0 0x40010000
    - cr 0x40010000
    - sr 0x40010004
    - int 0x40010010
    - count 0x40010020
    - match 0x40010024
    - prescale_rd 0x40010028
    - prescale_wr 0x40010028
    - reload0 0x40010050
    - reload1 0x40010054
    - reload2 0x40010058
    - reload3 0x4001005c
    - timer1 0x40010100
    - timer2 0x40010200
- AT91SAM9CN11 - 20130208
- EFM32G200F16 - 4.0.0
- LPC1102_4_v4 - 4
- M051_Series - 0.1
- M061 - 2.0.2.1

cr - ARM_Sample.timer0.cr

0x40010000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	11	10	9	8	7	6	5	4	3	2	1	0
s	rsvd2			idr		reload		rsvd1		trgent		rsvd		cntsrc				psc	mode			cnt		rst	en
0	0x0			0x0		0x0		0x0		0x0		0x0		0x0				0	0x0			0x0		0	0
0x00000000																									

Shareable direct link to this field

Base: B D H | Swap: Byte Word | Reset: Default

Control Register

Bits	Name	Access	Description
31	s	read-write	Starts and Stops the Timer / Counter
30:28	rsvd2	rsvd	Reserved
27:26	idr	read-write	Selects, if Reload Register number is incremented, decremented or not modified
25:24	reload	read-write	Select RELOAD Register n to reload Timer on condition

Powered by regvue v1.1.3

Direct Permalinks

regvue - Multi-SoC Example

github.jpl.nasa.gov/pages/regvue/regvue/v1.1.3/#/root/ARM_Sample/timer0/cr?field=idr&data=https://github.jpl.nasa.gov/regvue/demos/raw/main/various/all.json

Multi-SoC Example (v1.0) | Search Ctrl+K | GitHub

- ARM_Sample - 1.0
  - timer0 0x40010000
    - cr 0x40010000
    - sr
    - int
    - count
    - match
    - prescale_rd 0x40010028
    - prescale_wr 0x40010028
    - reload0 0x40010050
    - reload1 0x40010054
    - reload2 0x40010058
    - reload3 0x4001005c
    - timer1 0x40010100
    - timer2 0x40010200
- AT91SAM9CN11 - 20130208
- EFM32G200F16 - 4.0.0
- LPC1102_4_v4 - 4
- M051_Series - 0.1
- M061 - 2.0.2.1

cr - ARM_Sample.timer0.cr

0x40010000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
s	rsvd2			idr		reload		rsvd1		rgext		rsvd0		capedge		capsrc				cntsrc				psc	mode			cnt		rst	en
0	0x0			0x0								0x0		0x0										0	0x0			0x0		0	0
0x0000000																															

Reset: Default

Control Register

Bits	Name	Access	Description
31	s	read-write	Starts and Stops the Timer / Counter
30:28	rsvd2	rsvd	Reserved
27:26	idr	read-write	Selects, if Reload Register number is incremented, decremented or not modified
25:24	reload	read-write	Select RELOAD Register n to reload Timer on condition

Powered by regvue v1.1.3

Regvue application

Register path

Field

Register description JSON

JSON Input Format

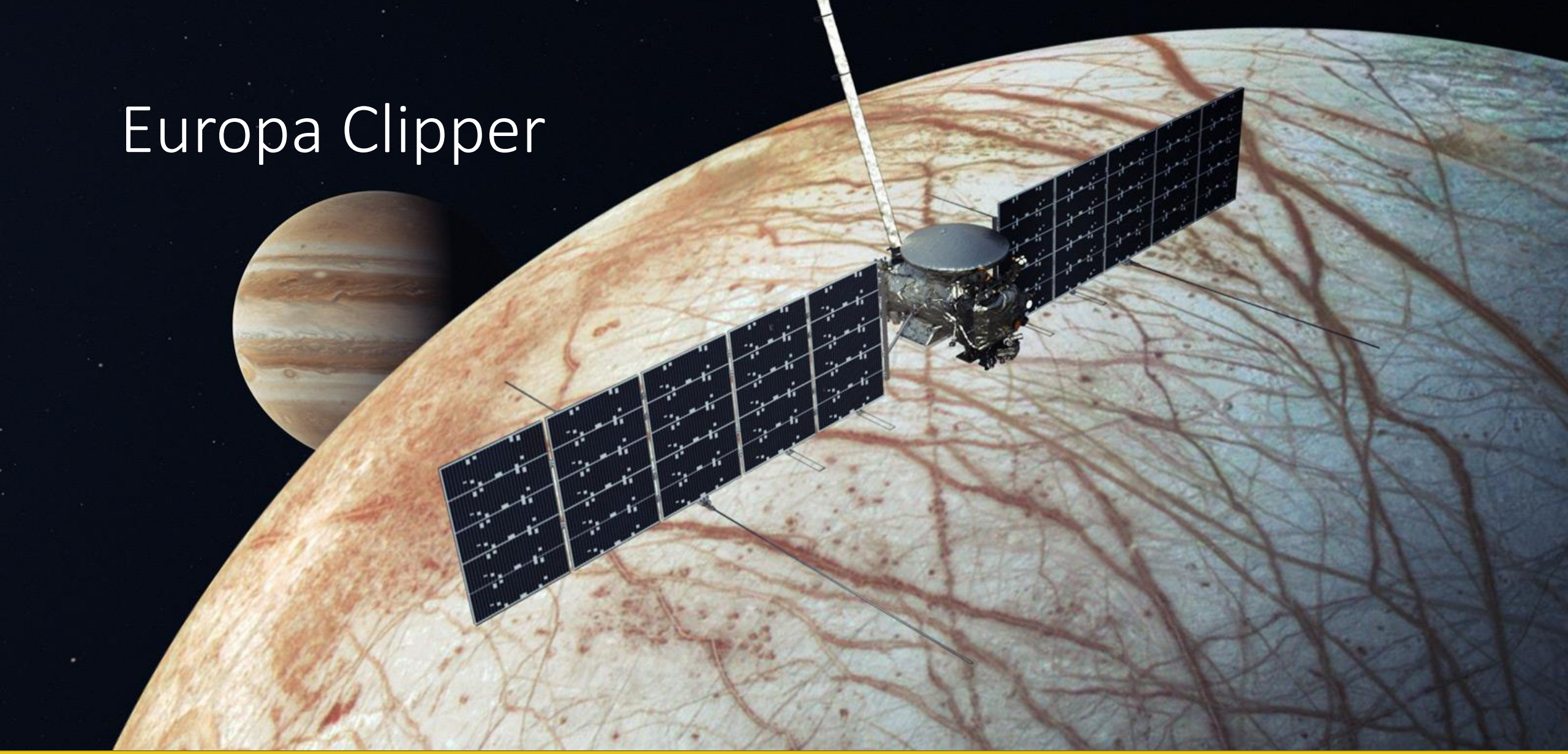
Schema Documentation

Table of Contents

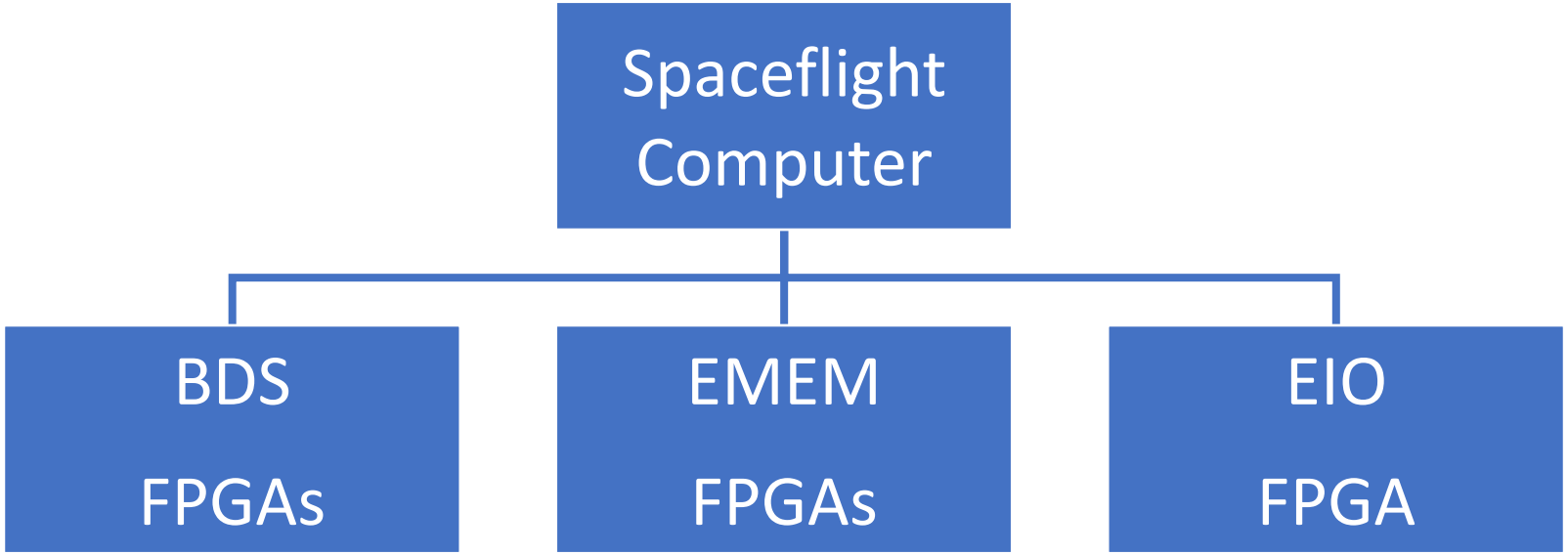
Example JSON

```
{
  "schema": { "name": "register-description-format", "version": "v1" },
  "root": {
    "desc": "Example Design", "version": "v1.0",
    "links": [ { "text": "GitHub", "href": "https://github.com/org/repo" } ],
    "children": [ "blk0" ]
  },
  "elements": {
    "blk0": {
      "type": "blk", "id": "blk0", "name": "blk0",
      "children": [ "blk0.reg0", "blk0.reg1" ]
    },
    "blk0.reg0": {
      "id": "blk0.reg0", "type": "reg", "name": "reg0", "offset": "0", "doc": "An example register.",
      "fields": [
        { "name": "rsvd", "lsb": 5, "nbits": 28, "access": "rw", "doc": "A reserved field." },
        { "name": "command", "lsb": 0, "nbits": 4, "access": "rw", "doc": "The command to execute." }
      ]
    },
    "blk0.reg1": {
      "type": "reg", "id": "blk0.reg1", "name": "reg1", "data_width": 16, "offset": "4",
      "fields": [
        { "name": "f0", "lsb": 0, "nbits": 16, "access": "ro", "reset": "0" }
      ]
    }
  }
}
```

Europa Clipper

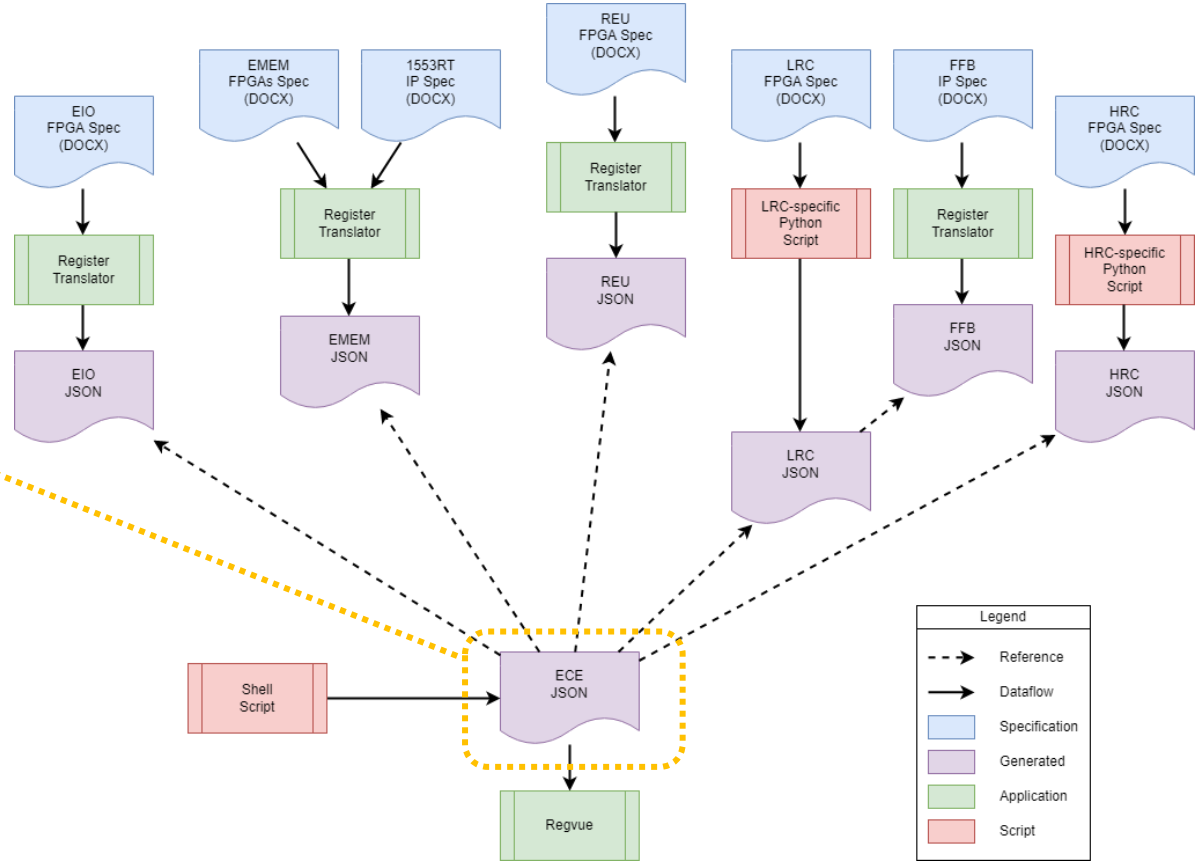


Europa Clipper Avionics

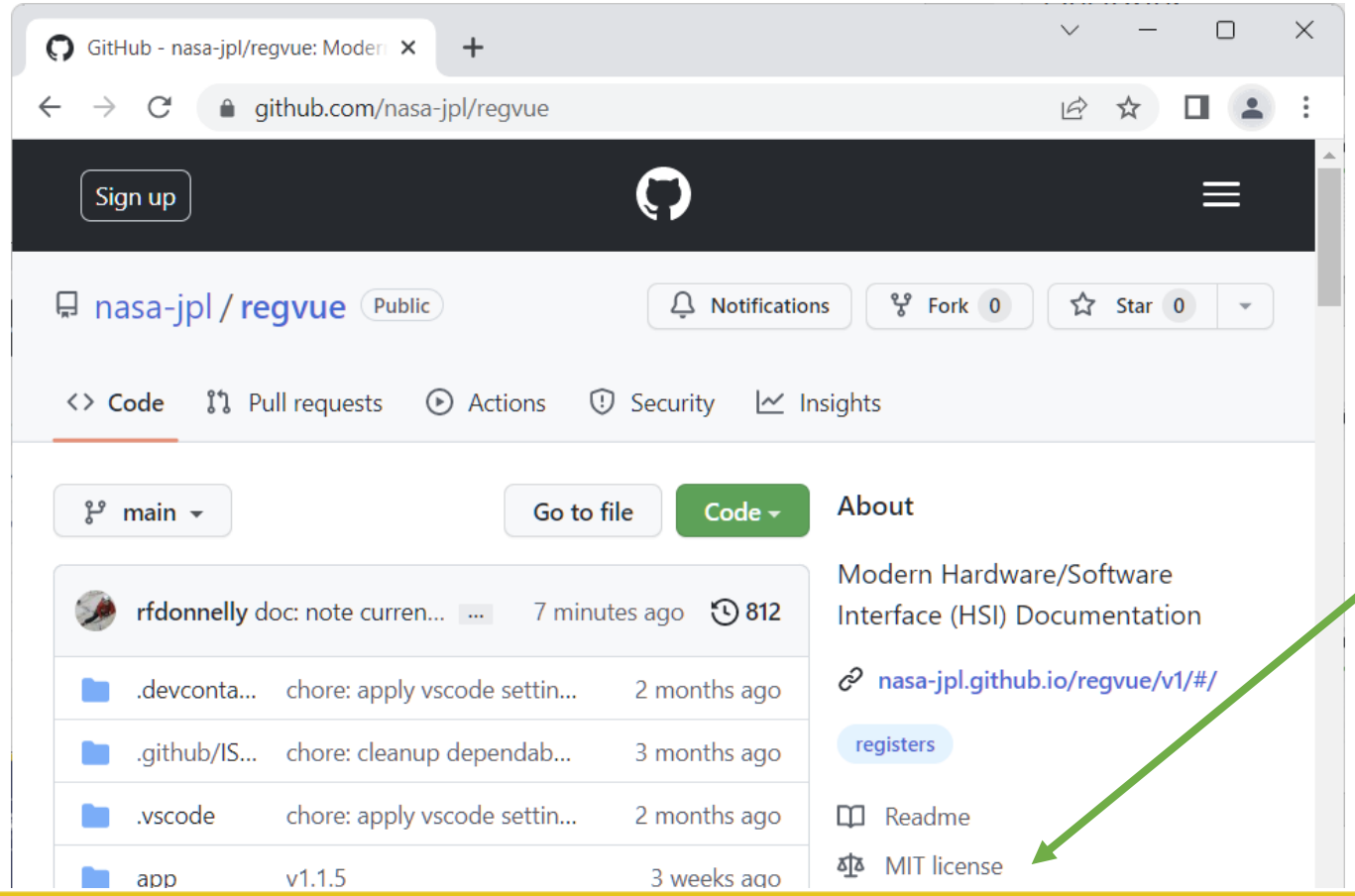


Europa Clipper Regvue JSON

- Convert several Microsoft Word documents to several Regvue JSONs
- Hand created top-level JSON includes other JSONs



Open Source @ https://github.com/nasa-jpl/regvue



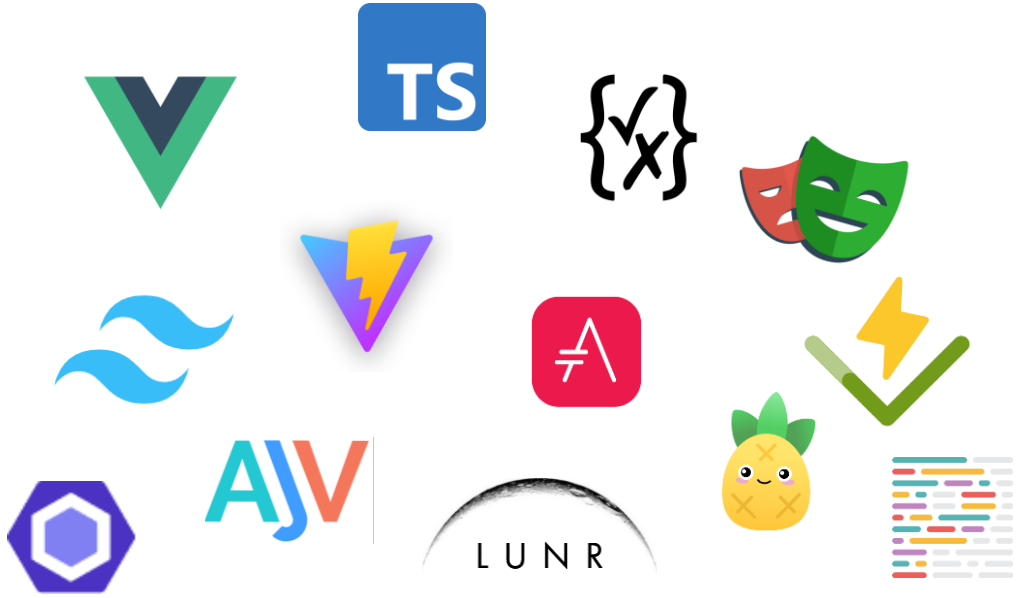
Permissive license

Summary

- Modern hardware/software interface documentation
- Increases efficiency
  - Direct link to specific design elements
  - Incremental search
- Reduces errors
  - Field/register decode/encode
- Open source

Questions

Open source technologies used



Funding provided by