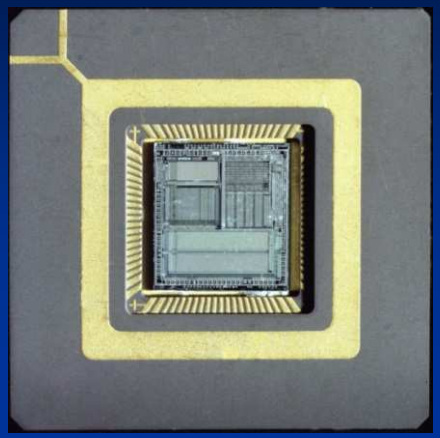


T42 – Transputer Design in FPGA

Year-Three Design Status Report

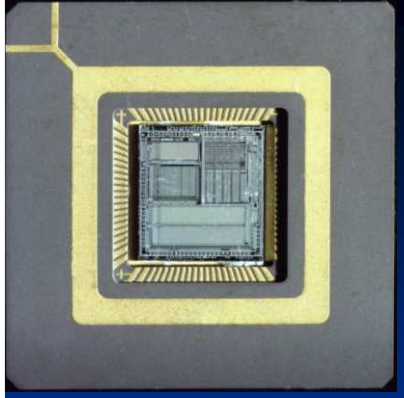
Uwe MIELKE [a] and Martin ZABEL [b] ,
in collaboration w/ Michael BRUESTLE [c]

[a] *Electronics Engineer, Dresden, Germany,* uwe.mielke@t-online.de

[b] *Institute of Computer Engineering, Technische Universität Dresden, Germany,* martin.zabel@tu-dresden.de

[c] *Electronics Engineer, Vienna, Austria,* michael_bruestle@yahoo.com

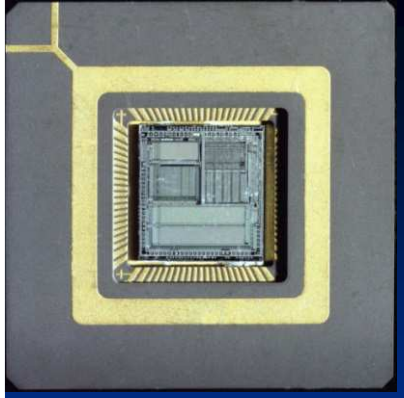
Communicating Process Architectures 2017



T42 in FPGA @ CPA 2016

Abstract:

Our IMS-T425 binary compatible Transputer design has so far taken over 300 design days. Up to last year minimal effort was spend for verification. Now a regression test bench has been brought in place, which is targeted to verify the design conformance after any changes. This T42 Transputer Verification Suite is based on a TVS-1 work from *Michael Bruestle* and compares the register output of 54 selected instructions versus a a true T425 golden reference for up to thousands of data samples. It helped already in T42 micro-code debugging and hardware refinement.



Agenda

(1) Review

(2) Regression Test Bench

(3) Transputer Verification Suite

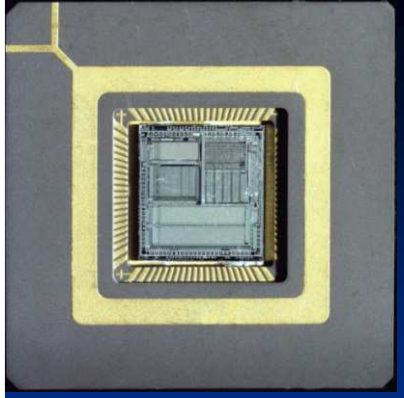
(4) TVS-1 Coverage

(5) Design Environment

(6) Achievements (2017)

(7) Outlook (Links ; Verification)

(8) Summary & Discussion



T42 in FPGA @ CPA 2014-16 (Review)

- 2-stage-pipeline for T42 working in 2014

  pre-fetch … is an autonomous FSM *)

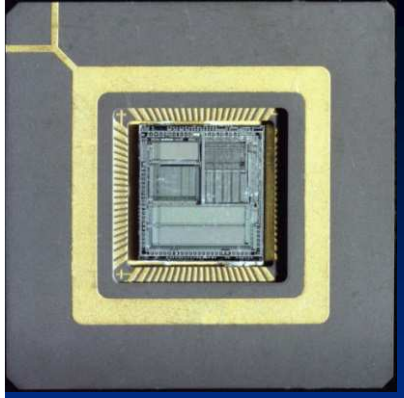
  1. **IF/ID** instruction fetch & decode
  2. **EX** execute (using a single or multiple clocks per instruction)

  memory read/write is part of execute *)

- micro code assembler & pre-fetch unit working in 2015
- system control & memory interface working in 2016
- >100 of 134 instructions (~600 lines of µCode) written

A lot of HW → How to prove everything is correct ?

*) not a pipeline stage

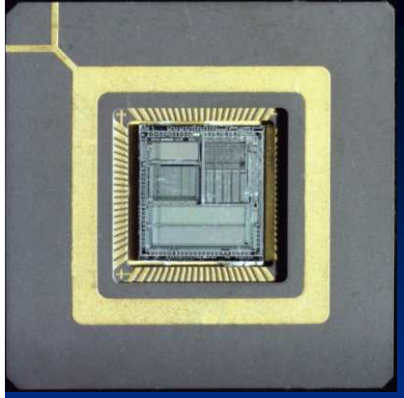


Regression Test Bench !

- Our T42-in-FPGA has taken over 300 design days up to now !
- A lot of HW ! But minimal effort was spend for verification yet.
- Visual inspection of simulations is cumbersome & erroneous !

- The purpose of a **Regression Test Bench** is to verify a design versus a (target) **Specification** (or a former stable state achieved).
- In our TVS-1 case the specification is the binary execution result of a part of the original T425 instruction set.
- The chosen instructions are the most important ones for a user program (compiler) & can be verified by a simple IUT algorithm.

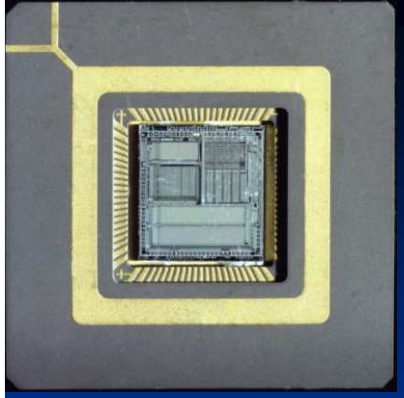
→ **Lesson learned**: verifications takes as much effort than design!



Transputer Verification Suite

- TVS-1*) uses a golden references based on real T425 outputs.
- Here: 54 IUT (instructions under test) with 1, 2 or 3 operands.
- IUT assembler code and sample set will be loaded into on-chip SRAMs before each run & comparison w/ golden reference file
- Adaption for VHDL simulation was required: reduction of basic sample set from 128*) to 32 values to achieve suitable run times.
- Basic sample set contains 32 signed integers (32bit): corner cases around MINT … Zero … MaxInt, several bit-pattern and single bit '1' and '0' values, some small and large integers.
- Permutations: if Areg & Breg loaded then 32x32=1024 sets used.

Info: *) TVS-1 was written by *Michael Bruestle* in 2010 to support software development & verification of the Transputer Emulator Project (*Gavin Crate*)



TVS-1 Coverage (Instructions)

TVS-1 covers 54 instructions:

- primary (3/16) ldc, adc, eqc, …
- arithm. logic (16/17) add, gt, xor, …
- long arithmetic (9/9) ladd, lsum, …
- indexing (5/8) bsub, wcnt, …
- error handling (2/8) ccnt1, csub0, …
- general (7/8) csngl, xword, …
- CRC and bits (5/5) bitcnt, …
- floating point (5/6) unpack, …
- ALT (2/12) alt, talt.

7 input files to meet different IUT requirements,
e.g. for arithmetic, shift, range check, FP, …

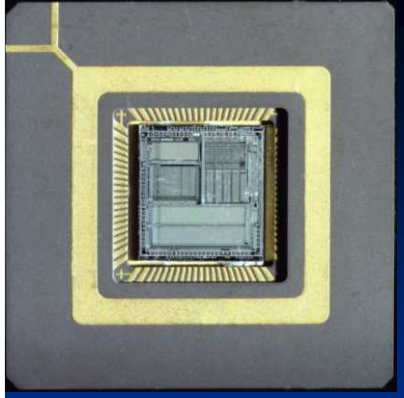
```
; load test

ldl  CREG
ldl  BREG
ldl  AREG

__IUT__

stl  AREG
stl  BREG
stl  CREG
testerr
stl  ERROR

; send result
```



TVS-1 Coverage (Samples)

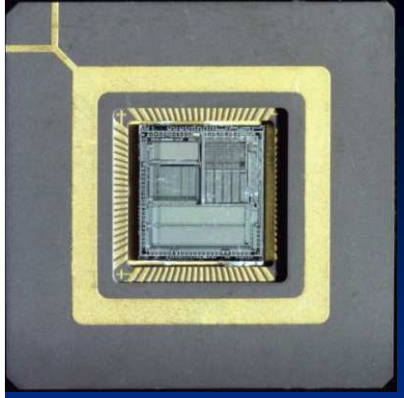
- **TVS-1 (original)**

TVS-1 original	input set	Extra Constants	Areg Values	Breg Values	Creg Values	NO. of TESTs	WORDs per SET	IN-WORDs
TEST.1	i32_1.bin		128	BBBBBBBB	CCCCCCCC	128	3	384
TEST.1.4	i32_1.bin	8	128	BBBBBBBB	CCCCCCCC	1.024	3	3072
TEST.2	i32_2.bin		128	128	CCCCCCCC	16.384	3	49152
TEST.3	i32_3.bin		128	128	8	131.072	3	393216
TEST.B	i32_B.bin		32	128	8	32.768	3	98304
TEST.F	i32_F.bin		64	BBBBBBBB	CCCCCCCC	64	3	192
TEST.P	i32_P.bin	14	8	72	14	112.896	**4**	451584
TEST.S	i32_S.bin		66	128	8	67.584	3	202752
						361.920		

- Original TVS-1 has ~360.000 tests (can be used over link only … may be later in case T42-in-FPGA is running on an FPGA board w/ PC connection)
- T42 TVS-1 will have ~25.000 tests (could be increased in case necessary)

TVS-1 Benefit:

- a TVS-1 run after (some) VHDL modifications will **verify** if the design still meets the specification ... or: if there is any (bad) impact from recent changes



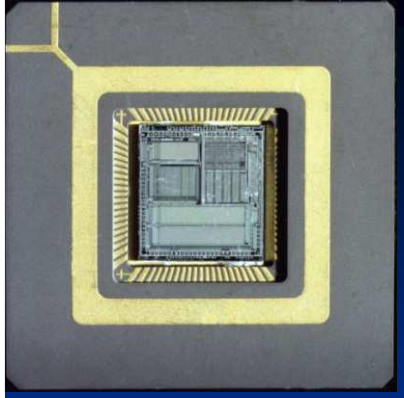
TVS-1 Coverage (Samples)

- TVS-1 for T42

TVS-1 (T-42)	input set	Extra Constants	Areg Values	Breg Values	Creg Values	NO. of TESTs	WORDs per SET	IN-WORDs
TEST.1	i32_1.bin		128	BBBBBBBB	CCCCCCCC	128	3	384
TEST.1.4	i32_1.bin	8	128	BBBBBBBB	CCCCCCCC	1.024	3	3072
TEST.2	i32_2.bin		32	32	CCCCCCCC	1.024	3	3072
TEST.3	i32_3.bin		32	32	8	8.192	3	24576
TEST.B	i32_B.bin		32	32	2	2.048	3	6144
TEST.F	i32_F.bin		64	BBBBBBBB	CCCCCCCC	64	3	192
TEST.P	i32_P.bin	*10*	*8*	*10*	*10*	*8.000*	**4**	32000
TEST.S	i32_S.bin		32	32	4	4.096	3	12288
						24.576		

Output example:

```
  prep_iut.BAT: IUT is   ADC   prepared @ 11.08.2017 11:46:43,65
tb_07_tvs1.vhd: simulation started...
tb_07_tvs1.vhd: simulation Ok. - 4096.word - end of ..\sim\tb_07\golden_reference.mem reached.
  ghdl_sim.BAT: IUT is   ADC   finished @ 11.08.2017 11:47:11,96
------------------------------------------------------------------------
  prep_iut.BAT: IUT is   ADD   prepared @ 12.08.2017 12:05:46,48
tb_07_tvs1.vhd: simulation started...
tb_07_tvs1.vhd: simulation Ok. - 16384.word - end of ../sim/tb_07/golden_reference.mem reached.
  modelsim.BAT: IUT is   ADD   finished @ 12.08.2017 12:13:25,95
```

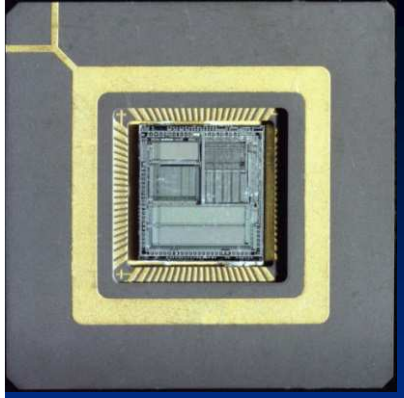


Design Environment → SVN

- Our T42-in-FPGA design environment was Xilinx ISE 14.7
- ISE is **not supported** anymore by Xilinx (now used is: Vivado)
- ISE has still some bugs … e.g. handling of "X" values …

- Therefore a change was required to be more „state of the art"
- **Now:** 3 different **Simulators** can be used (more is better!) :
  - GHDL v0.33 (free) from *Tristan Gingold* … + GTKWave Viewer
  - Mentor ModelSim – as part of ALTERA FPGA lite 16.1
  - Xilinx iSim – as part of ISE 14.7
- 2 of these simulators can be scripted (automated :-) nicely

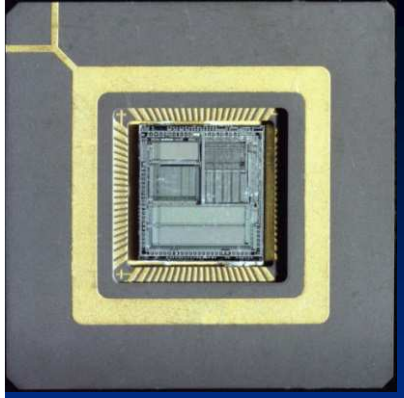
→ **Our improved Design Environment looks "CMOS alike" now ;-)**



T42 Achievements 2017

- Memory preparations, e.g. 1024x128bit_ucrom … … Nov.2016
- TVS-1 regression test bench adaption started … … Nov.2016
- clean up: fix simulation warnings about "X" values … Jan.2017
- 1st TVS-1 instruction tests running (ADC, …) … Feb.2017
- all 54 TVS-1 instructions prepared, 50% running … Apr.2017
- data path + uCROM review (→ more instructions) … May.2017
- full design environment reorganization (→ SVN) … Jul.2017
- 42 of 54 TVS-1 instructions verified … … Aug.2017

Note: switch to 1024x128bit_ucrom, e.g. merging of all available instructions & routines in about ~600 lines of microcode will be done in autumn …



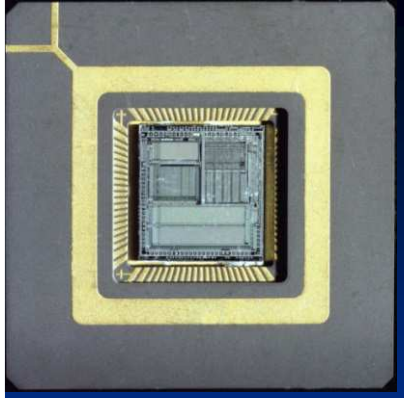
Outlook 2017Q4+ Verification

still a lot of work to do: *(partially pre-prepared already)*

- IUTs to debug: IN, OUT, ALTs, PARs, CRCs, …
- IUTs in completion: MULs, DIVs, FP support
- there are still ~80 instructions w/o verification concept yet !
- scheduler µCode to debug: start next process, enqueue, dequeue …
- scheduler µCode completion: HW event interaction w/ timer, links, boot, peek, poke, Halt-on-Error, analyse, …
- verification pending: Timer & time slice, …
- reverse engineering & verification: Links + control logic *(in work)*

by help of TUD & POC library: *(pre-prepared)*

- connect external memory: cache & DDR-RAM controller

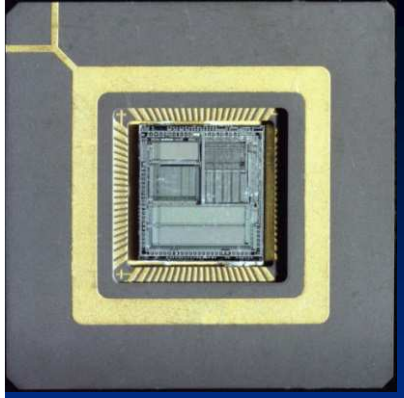


Outlook 2017Q4+ Links …

CHANNEL ADDRESSING by different Instructions

- IN , OUT
  - Areg = unsigned **message length** in byte → 2nd push onto link stack via Zbus
  - Breg = pointer to the channel
  - Creg = **pointer to the message** → 1st push onto link stack via Zbus
- TESTHARDCHAN
  - Areg = selected channel
  - Breg = **data** (push into top of Link-DMA-Stack)
  - Areg' := DBuffReg (read from bottom of Link-DMA-Stack)
  - Breg' := Creg
  - Iptr' := next instr.
- Link-DMA-Stack …
  - CountReg' := Breg
  - PtrReg' := CountReg
  - DBuffReg' := PtrReg

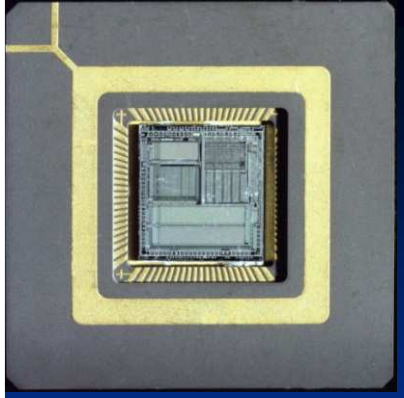
(Zbus) (Ubus)
Breg → CountReg → PtrReg → DBuffReg → Areg



T42 Summary

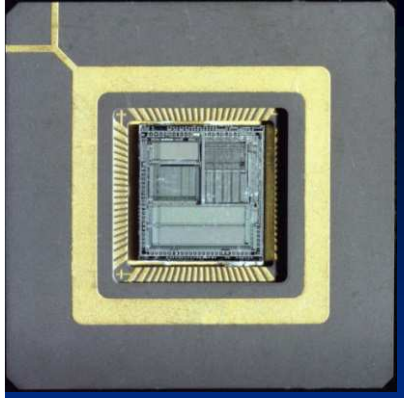
1. "patch work" Transputer implementation is available
   - integer CPU & timer, (links … in work)
   - simple memory path („on-chip")
   - by TUD pre-prepared & tested: generic cache + DDR-RAM controller
2. semi automated micro code generation possible
   - one update run takes less than 1 minute
3. regression test bench for verification in place
   - one full verification run (e.g. 42 instructions) takes < 1 hour
4. already >100 (of 134) instructions in µCode written
   - ~70 tested & used (e.g. primary & initialisation instructions, move, lend)
     - ~40 instructions proven correct by TVS-1

→ best conditions to continue & finish the whole design!

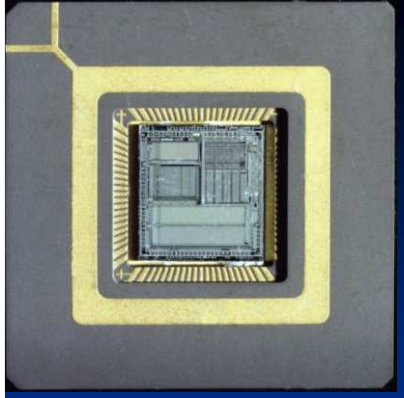


Time for Questions …

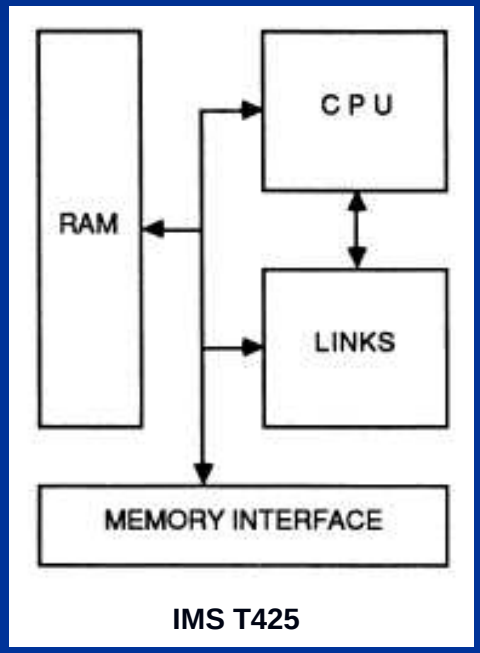
… we're on the right way



BACKUP

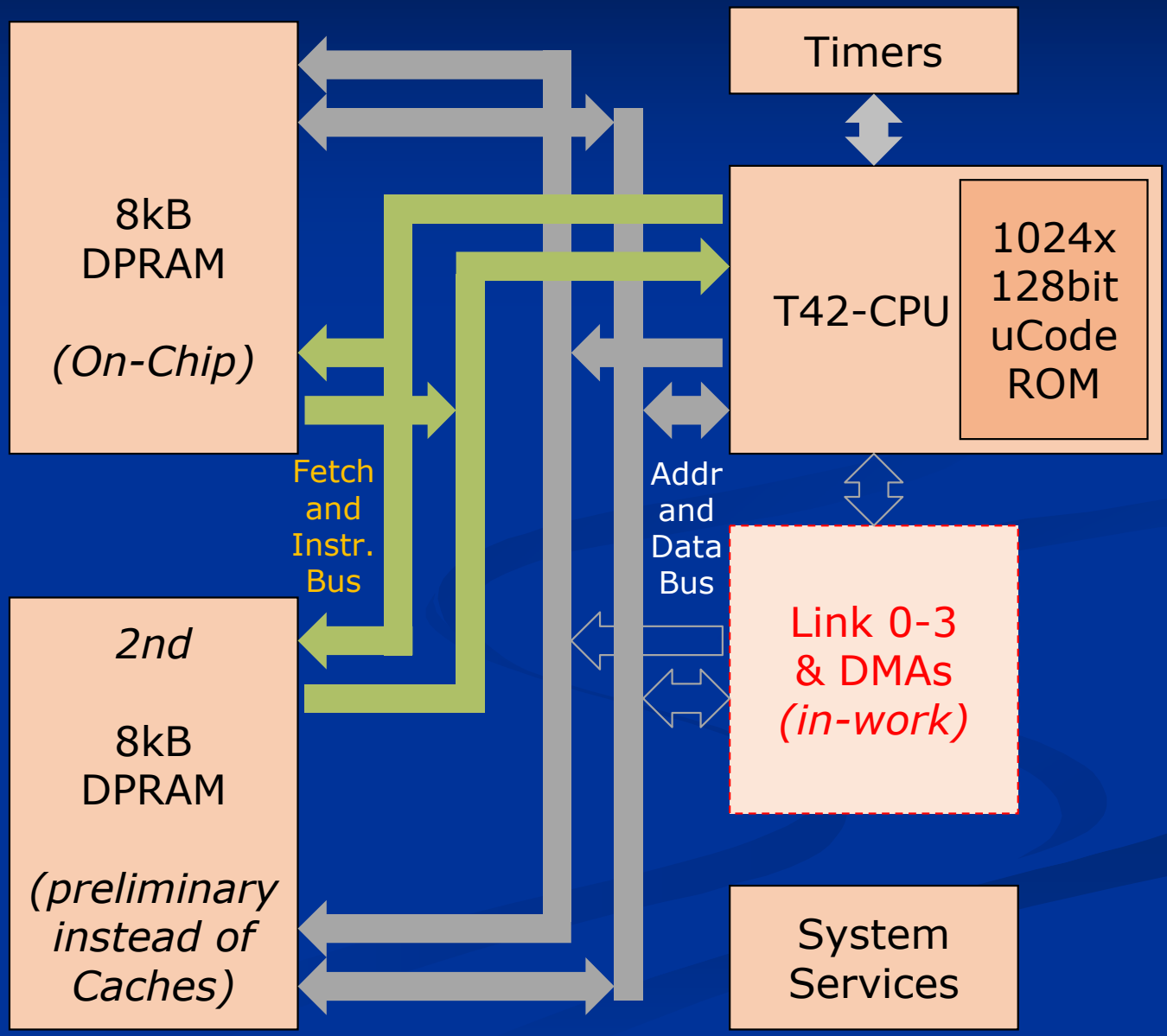


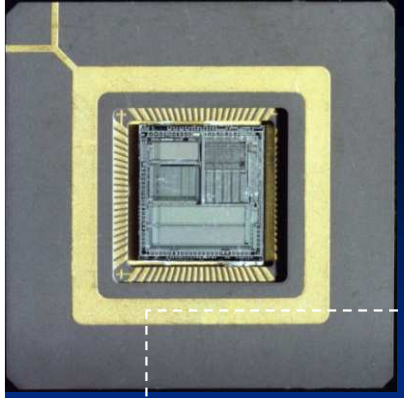
T42 Schematic



IMS T425

Note: T425 µCROM is **742x118bit** (~90kBit) investigation done in Nov.2016 by *Gavin Crate*





Remark: Blocks in red still N/A.

T42 VHDL Top View

t42cpu_all_top (structural)

SysPath:
- SysCtrl, Sbits, Timer, SysService

t42_cpu_constpkg
t42_cpu_functpkg

Ctrl2Data (structural) ← *Pipeline*

CtrlPath:
- uCodeROM
- Idecode
- Oreg *(pipe)*
- Iptr *(+Inc)*
- PreFetch

DataPath:
- ABCDEreg
- ALU *X+Y=Z*
- Wptr
- Pointers
- ConstBox
- ByteAlign

LinkPath:
- Sync, ChOut, ChIn, ChEvent, Ifos

MemPath:
- MemIF
- MemMain (dpram2kx32)

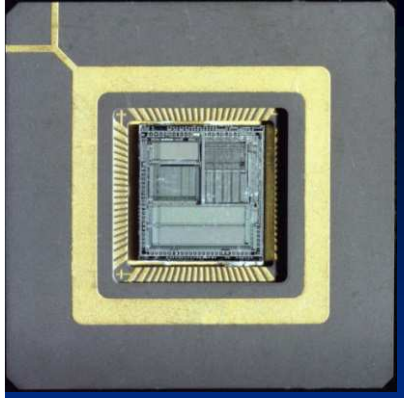
preliminary…
…instead of cache
- DummyCache (dpram2kx32)

available+tested:
- CacheCtrl *(TUD)*
- DDRCtrl *(TUD)*

Target Board No.3 99$
Digilent Arty
MemDDR3
(128Mx16 on board)
XC7A35T

Target Board No.2 199$
Digilent ATLYS
MemDDR2
(64Mx16 on board)
XC6LX45

Target Board No.1 89$
Avnet Micro Board
MemLPDDR
(32Mx16 on board
XC6LX9



INMOS Patent Research

- Inmos Innovation and Patents: http://www.petritzfoundation.org/?mdocs-file=968
  The "In" in Inmos Stands for Innovation ; James R. Adams, PhD

- US-Pat-4704678 - INMOS 26Nov1982 https://patents.google.com/patent/US4704678A/en
  Function set for a microcomputer
- US-Pat-4724517 - INMOS 26Nov1982 https://patents.google.com/patent/US4724517A/en
  Microcomputer with prefixing functions
- US-Pat-4758948 - INMOS 19Jul1988 https://patents.google.com/patent/US4758948A/en
  Microcomputer
- US-Pat-4989133 – INMOS 29Jan1991 https://patents.google.com/patent/US4989133A/en
  System for executing time dependent processes
- US-Pat-4783734 – INMOS 08Nov1988 https://patents.google.com/patent/US4783734A/en
  Computer with variable length process communication
- US-Pat-4794526 – INMOS 27Dec1988 https://patents.google.com/patent/US4794526A/en
  Microcomputer with priority scheduling