

Needs Within and Above the Exascale Program

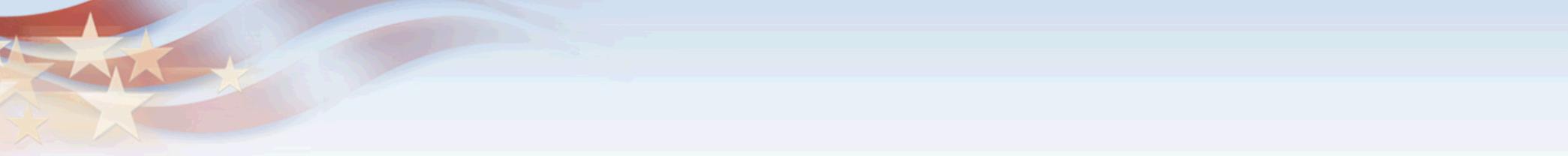
Make it easier to evaluate and implement new ideas at all levels, and to maximize the opportunities presented in developing new high-end systems



Making New Development Better

Reduce effort and complexity, improve results:

- Make architectural experiments easier
- Develop ways so that full multi-discipline co-design is how things are done.
- Find ways to make development programs more innovative and less risky for developing companies.
Invest in and support paths to bring users and customers to new capabilities



Need Better/Easier Experimentation?

- Large system companies do most of their own development work but are limited by the large costs and time it takes to do really new work to the needed depth such that companies are willing to take new paths
- Universities get lots of new ideas. But: very complex, time/labor consuming, students interest in simply 'being done,'.... Hard to have multi-person efforts involving multiple technologies
- Programs where companies support new efforts (e.g.: parallelism), but limited number of people make the innovations and get the benefit of working 'at the edge.' Programs often do not consider full system level impact

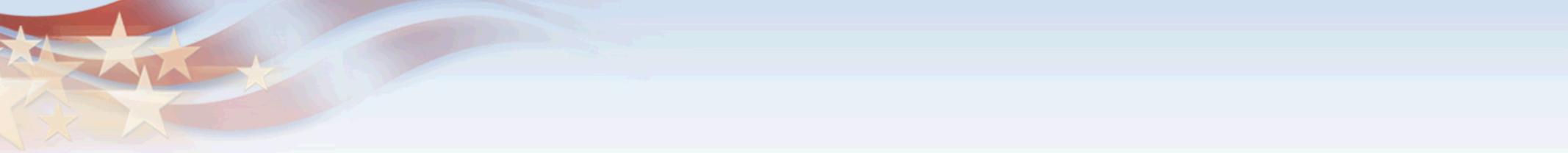


Technical Support in Industry

- Most high-technology companies provide extensive support for users of their components or capabilities
- Companies like: Altera, Xilinx (FPGAs); Micron, Samsung (memory components); Mentor, AutoCAD ...
- Support includes development systems, inexpensive CAD tools, example designs, field-application engineers, technical documentation, whitepapers, etc., etc.

Customers hear:

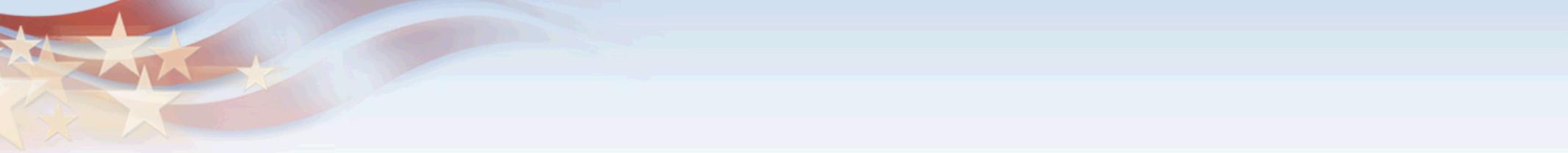
“Here are multiple easy ways to use our parts—tools and expertise—including help for the portions that are needed but not the focus of your effort”



Supporting Experimentation

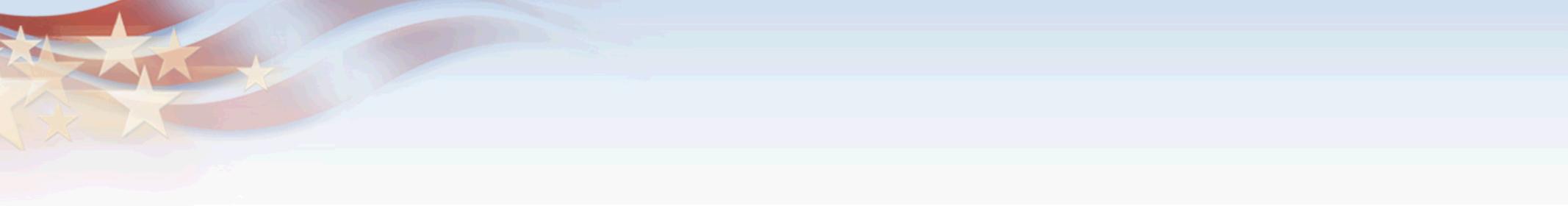
Have an institution (group? program?) whose purpose is to support experimentation

- Provide expertise and development support to users for system components that they can't reasonably do themselves and are not the objective of the experiment
- Can charge reasonable fees, but *make things possible* as some ideas take too much time, resources, or expertise—but are more than worthwhile doing
- Plan the results be open source (or increased cost if not), so growing the capability
- Work out IP issues



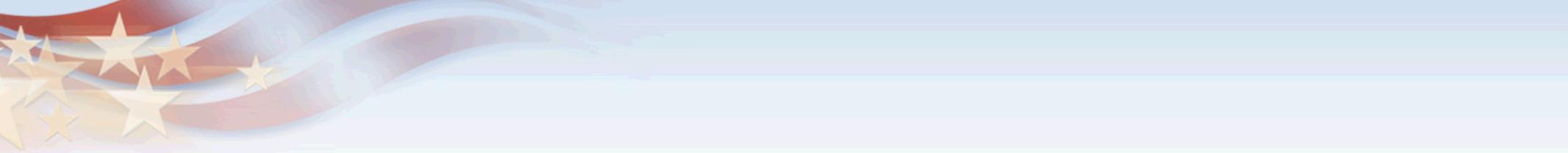
Support Experimentation

- Have prebuilt systems & CPUs with lots of knobs (LKS)
- Have multiple scalable networks with LKS
- Have multiple memory architectures with LKS
- Have multiple IO systems with LKS
- Make available development people do needed work that is not within the expertise of the experimenter
- Provide many kinds of application codes with help in changing to support desired experiments
- Provide multiple compilers and libs, again with support for changes
- **Make all components function in a common framework**



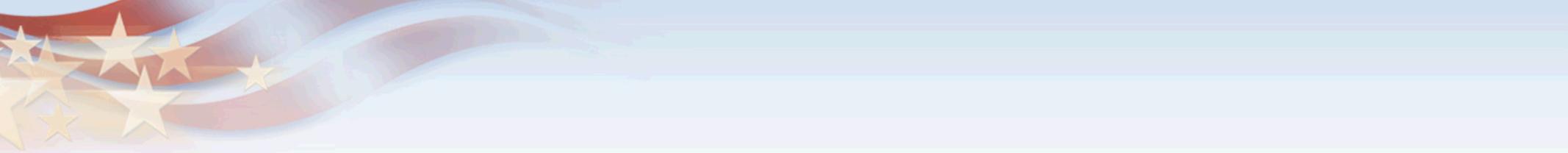
Co-Design

Making it work



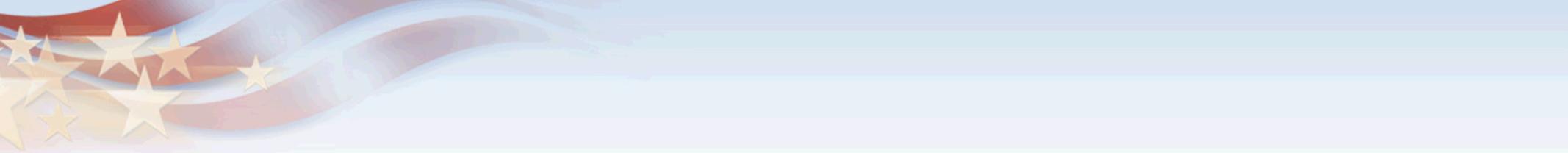
Co-Design Is Not:

- Having a meeting upfront, and then parallel working groups that then mash separate results together into a compromise implementation
- Having monthly meetings—and then mostly ignoring any changed direction (except for complaints)



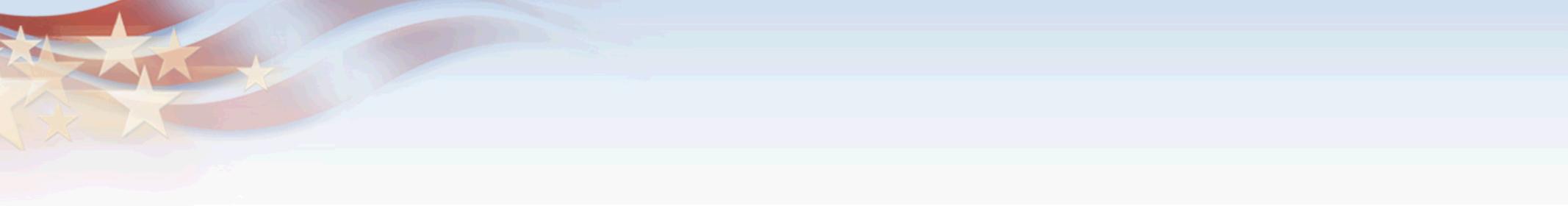
Co-Design Should Be:

- Ongoing interaction
- Groups offering new ideas to other groups—all the time
- Being aware of the needs and requirements of other groups and asking for and offering alternatives for what you are doing
- Developing ideas together, both at high levels and within subgroups
- Making decisions across multiple groups
- Managing cooperatively
- Making compromises between groups *before* work is done, and ongoing



Making Co-Design Happen

- Frequent interaction: meetings, calls, emails because they are useful, not because they are required
- Input, feedback, general consideration at decision points
- Be willing to have your plans affected by other groups (within set parameters (e.g.: no change in A,B,C after this date!))
- Reporting on expected inputs to/from other groups
- Keep cooperative efforts visible to all; support communication with wikis, data bases, publication of open issues and questions, ...
- **Build in the expectation of cooperation and interaction at all project levels**



Better Support for Innovation in System Development

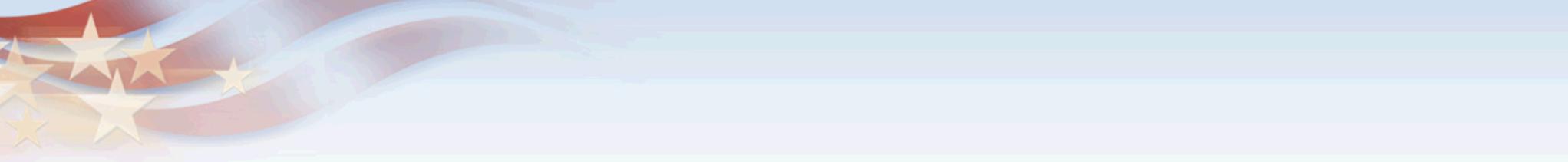
Reduce risk,
Keep more new ideas
Plan to support change



Need for Innovation support

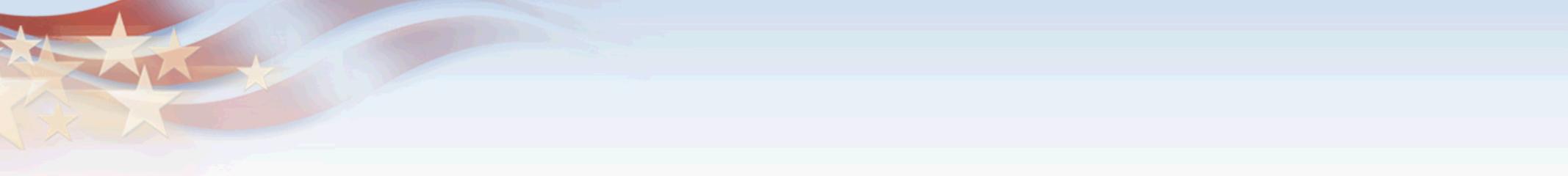
Exascale program currently has multiple research efforts going; planning on a system company picking up the results and making commercial systems

- Generally one-off systems can not be justified on expected return-on-investment
- But the promise of commercial sales also greatly increases the risk seen by the developing company and increases and complicates the effort.
- Hard to insure that the new ideas and work done in getting to a system contract award will be actually used in resulting system(s)



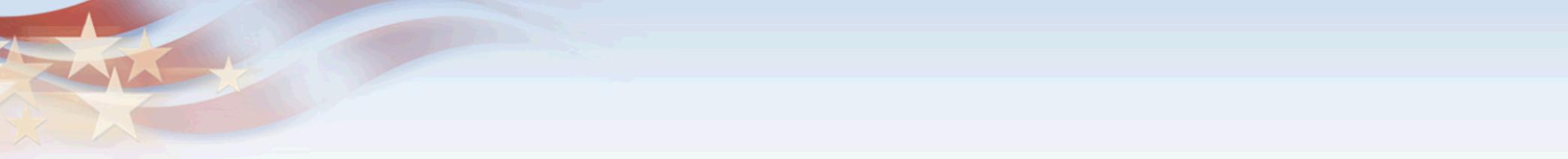
Need for Innovation Support

- The more a new system is different than current systems the more has to be done to show the effort is worthwhile and to find ways to show believable ROI
- The more a new system is different than current systems the harder it is to understand the reasons for the changes and to justify new work rather than 'doing it the old way.' Engineers (and managers, and accountants, and marketing, and...) understand best what they did last :-)
- The more a new system is different than current systems the more new things that must be done and paid for
- The more a new system is different than current systems the harder it is to bring users up to speed



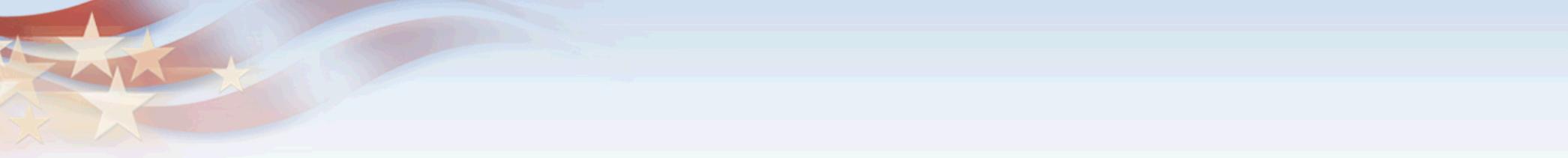
So...

- Ensure that the benefits of what is new and different is understood by all
- Make design changes visible and ensure justification
- Include outside, but fully knowledgeable, people in the decision path and to get input on tradeoffs
- Have back-up plans and make the possibility of using them real—but not the focus of the development (where people ‘know’ that a backup plan will be used, so little pressure on actually making an innovative idea happen)
- Build in a support roadmap that enables users and customers to reasonably get to the envisioned future; make the roadmap happen



Help Support and Justify Innovations

- Have outside oversight group funded to interact with the system company, to consult, and to do some level of support for system development. (Provide input and justification for innovations: 'The benefit is actually worth the effort and cost')
- Oversight group built from previous work/before the contract efforts, so familiar with the what and why of what is proposed
- Oversight group is constrained with respect to IP issues



Support Getting to Tomorrow

- Part of bringing real innovation to market is supporting all the current users who are not using the new capabilities. Otherwise the risks are too high, both for the development companies and for the users
- The cost of supporting current users and getting them up to the state of using new capabilities must be part of development
- Implement a 'getting to the future' plan for customers and users, including the resources needed to accomplish it, and make that an integral part of development
 - ◆ Virtual systems and ...
 - ◆ Transition tools and ...